

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

Lecture - 52
Toda's Theorem: Part 1

(Refer Slide Time: 00:15)

Lecture 52 - Toda's Theorem
23 September 2021 00:15

[loosely based on notes from Prabhakar Harsha
& Arora-Barak textbook]

We've seen that #P is at least as powerful as
P, NP, BPP, RP.

→ We know that $P^{NP} \subseteq \Sigma_2$.

→ What if we replace NP with #P?

→ How powerful is $P^{\#P}$?

... $P^{\#P}$

NPTEL

Hello and welcome to lecture 52 of the course computational complexity. We have been seeing the power and complexity of counting over the past few lectures. In this lecture we will see Toda's theorem which was a surprising result. And this stated that the entire polynomial hierarchy can be captured with the sharp P by sharp P. So, let us see the definition, let us see the statement.

So, the statement is that polynomial hierarchy is contained in P with a sharp P oracle. So, if there is a sharp P oracle which can answer queries from the function plus sharp P. A polynomial time machine for deterministic polynomial time machine with a sharp P oracle can simulate a polynomial hierarchy any level of polynomial hierarchy. So, why is this interesting?

So, we know we have already seen that a sharp P is at least as powerful as P, NP, BPP, RP and so on VP. Because if we can count, we can decide whether there is at least 1 accepting path or 0 accepting path. So, it is you can decide you can make it an NP language and

certainly you can decide whether at least two thirds of the paths are accepting or at most one third.

So, you can decide BPP and similarly you can decide PP, RP any of the randomized classes. Another thing that we know is P with an NP oracle, is contained in sigma 2 because P with an NP oracle can simulate both NP as well as co NP. So, it is bigger than or seems to be bigger than NP but both NP and co NP are contained in sigma 2. So, P to the NP is contained in sigma 2 in fact NP to the NP we saw that it is con it is equal to sigma 2.

So, now the question is, when we had P with an NP oracle, we got some language that is we got a class that is bigger than NP. Now what if we replace NP with sharp P here? We know sharpie can simulate NP but from sigma 2 in the right-hand side how far do we go.

(Refer Slide Time: 02:39)

The slide contains handwritten notes in blue and red ink on a lined background. At the top right is the NPTEL logo. The notes are as follows:

- What if we replace NP with #P?
- How powerful is $P^{\#P}$?
- Theorem (Toda '91): $PH \subseteq P^{\#P}$.
- We will see that Σ_k -SAT can be simulated by a det poly time machine with a #P oracle. That too, we will only ask me queries to the oracle.

In the bottom right corner, there is a small video inset showing a man with glasses and a green shirt speaking.

So, that is the question, and the answer as I already stated is that the entire polynomial hierarchy can be simulated with P, with access to a sharp P oracle. So, what we will see is that any language in sharp P sorry, polynomial hierarchy. So, the kth level of polynomial hierarchy we can call it sigma k. So, and the complete problem of that class we already mentioned is Σ_k SAT, with k quantifiers.

So, let me in this lecture let me just call it sigma k SAT which is the same thing and just calling with a different name. So, it is satisfiability with k quantifiers k alternating quantifiers, this is the complete problem for the kth level of sigma k, kth level of the polynomial hierarchy sigma k. So, Π_k will have the negation complement problem of this.

We will show that this language can be simulated by a polynomial time deterministic polynomial time Turing machine with access to a sharp P oracle. So, which is what we are saying here.

(Refer Slide Time: 03:49)

We can do the following operations.

Suppose ϕ has m and ϕ' has m' sat. assignments

→ Consider $\phi(x) \wedge \phi'(y)$.

This has $m \cdot m'$ satisfying assignments.

→ $[\phi(x) \wedge z] \vee [\phi(x) \wedge \bar{z}]$

This has $m + m'$ sat. assignments.

→ Can build a formula that has $c > 0$ satisfying assignments.

The image shows a whiteboard with handwritten text in green ink. In the top right corner, there is a circular logo with a star and the text 'NPTEL'. In the bottom right corner, there is a small video inset showing a man with glasses and a green shirt speaking.

And not only that a sharp P oracle it is a Turing reduction, you could ask multiple queries. What we will show is that this oracle we can make with, it is enough to just ask one query to this oracle. All we need to do is just ask one query not even multiple queries. So, now let us see the proof. So, I will try to explain the high-level details and then go into the final details of the proof.

So, I will be happy and the proof is kind of long it will be spread over two lectures. I will be happy so please try to understand the high-level picture of the proof first and then try to understand the lower-level details. So, I guess that will give a better understanding of the entire what is going on. So, let me just let us just first see some operations. So, suppose there is a formula ϕ that has m satisfying assignments.

So, this is and ϕ' has m' satisfying assignments. So, the objective is to show that if you have ϕ with m satisfying assignments and ϕ' with m' satisfying assignments you can do some kind of arithmetic in this sharp P world. So, you can for instance you can produce a formula, so consider this AND so you instantiate ϕ with the variable set x and ϕ' with a different variable set y .

And take the AND of these two very these 2 formulas. Then any satisfying assignment of this ϕ and ϕ' has to have it could be any of the satisfying assignments of ϕ and any of the satisfying assignments of ϕ' . So, x and y are independent set, so there is no dependency across. So, it could be any one of the m satisfying assignments of ϕ multiplied by and any of the m' satisfying assignments of ϕ' .

So, the number of satisfying assignments of this formula is m times m' . So, we are trying to show that we can do such arithmetic if you are given a formula with m and formula with m' then you can do something to produce a formula with m times m' . And similarly, if you can do this, you can instantiate ϕ and ϕ' with the same set of variables even if it is the same set of variables.

What you can do is? You can do this kind of thing where you and ϕ with z or ϕ' with z complement. So, ϕ' is another and the formula so either z has to be true or false, when z is true it does not matter what ϕ is what x are chosen as far as this the first clause is always true. But when z is z is true then the second clause, we need ϕ' to be satisfied.

So, any satisfying assignment of ϕ' have to be will is necessary to satisfy the second class. Similarly, when z is false the second clause is automatically true. But any satisfying assignment of ϕ is required to satisfy the first loss. So, when z is true then the second cross ϕ' needs to be satisfied when z is false the ϕ needs to be satisfied. So, this has $m + m'$ satisfying assignments.

(Refer Slide Time: 07:28)



This has $m+1$ satisfying assignments.

→ Can build a formula that has $c > 0$ satisfying assignments.


x has 1 sat. assignment.

$[x \wedge z] \vee [\bar{x} \wedge \bar{z}]$ has 2.

By addition, multiplication we can get any constant.

We will prove Cook's theorem in two parts.

more (ch 1 of 1 ...)



And one more point is that we can build a formula that has a constant number of satisfying assignments for any constant. So, let me just show some simple things so just a single variable formula x has one satisfying assignment, something like this x and z or x bar x complement and z complement has 2 like both 0 1 and 1 0 works are the 2 satisfying assignments. And now you have two formulas that have 1 and 2 satisfying assignments.

Now you can multiply to get 2 and 2 to get 4, 4 and 2 to get 8. And then you can get all the powers of 2 and then you can add to get any natural any positive integer or any natural number. So, even if you have a formula 5 with an unspecified number of satisfying assignments now you can get another formula with let us say ϕ hat x or m number of satisfying assignments. We could have another formula with $m + 10$ satisfying assignments.

Because we can make another, we can construct another formula that has exactly 10 satisfying assignments and then you can do this add operation. So, we can get any constant, so you can we can do $5 + 1$ or we can do an $+ 1$ operation. So, to if I had m satisfying assignments then you can build a formula that has $m + 1$. So, this kind of arithmetic with the number of satisfying assignments you can do.

So, what can we do? Into multiplication we can do addition and we can construct all positive integers.

(Refer Slide Time: 09:11)



We will prove Toda's theorem in two parts.

Parity

$$\oplus \text{SAT} = \{ \phi \mid \phi \text{ has an odd no. of sat. assignments} \}$$

Clearly $\oplus \text{SAT} \in P^{\#P}$.

Main

Theorem: (hard. reduction from $\Sigma_k \text{SAT}$ to $\oplus \text{SAT}$)

Let $k, m > 0$. There is a probabilistic poly. time reduction A that given a $\Sigma_k \text{SAT}$ instance Ψ , outputs a $\oplus \text{SAT}$ instance $A(\Psi)$.



So, we can do add positive integers. So, coming back to Toda's theorem we will do Toda's theorem in two parts. So, the first part will be covered in this lecture and second part in the next lecture. So, consider the language called sharp SAT sorry parity SAT. So, parity SAT is simply asking, so this plus with the circle plus inside the circle is called parity. It is simply asking if the given formula or it is a class of language or class of formulas that have exactly an odd number of satisfying assignments.

So, any formula it has a some number of satisfying assignments that number it could be 0 could be 1 could be 2 power n it could be odd or even. So, whatever that has odd it is in disparity SAT and clearly parity SAT is contained. So, it is a language, because it is everything is odd or even. So, parity SAT is a language it is not a promise problem or anything. It is not a function as well.

So, anything so clearly parity SAT is contained in P to the sharp P, P with a sharp P oracle. Because if you can count the number of satisfying assignments all you need to do is just decide whether the count is odd or even. So, clearly parity set is in P to the sharp P. So, again our goal was to show that polynomial hierarchy is in P to the sharp p, but this is something that we are just saying parity chart parity SAT is in P to the sharp P.

And what we will do is to show that any quantified Boolean formula with k levels of quantifiers. So, again this language is called sigma k SAT, so which has k quantifier starting with an existential quantifier. We will reduce it to sharp parity SAT and this reduction will be

a randomized reduction. So, if given k , there is a given k where k is the same and given in number m there is a randomized reduction, randomized polynomial time reduction.

That given a Σ_k instance ψ it outputs a parity SAT instance $A\psi$. So, you can think of A being the reduction, such that the Σ_k it has quantifiers and eventually it has a truth value true or false. If Σ_k is true or if the Σ_k instance ψ is true then the output instance $A\psi$ is very likely to be in parity SAT to be a sense instance of parity SAT meaning it is very likely to be very likely to have an odd number of satisfying assignments.

And notice that $A\psi$ is just a Boolean formula. It does not have quantifiers or anything. And then we are just checking how many satisfying assignments does it happen. And similarly, when ψ is false the Σ_k case at instance ψ is false then $A\psi$ will be very unlikely to be in a parity SAT. And the probability is like with probability $1 - 1/2$ problem with very high probability it is likely to be this correspondence is maintained.

So, it is not a deterministic full proof correspondence like a sense here is always mapped to a sense and here. But if ψ is true then we are very likely to have $A\psi$ being a P sense of parity SAT and if ψ is false, we are very likely to have $A\psi$ being a no instance of parity SAT. Another way to see this is that, so the left-hand side we have a Σ_k case SAT which is a like arbitrary language from the polynomial hierarchy.

So, polynomial hierarchy is contained in BPP with access to a parity P oracle so you can think of a parity P oracle instead of an oracle that counts the number of satisfying assignments. And oracle that just tells you, whether the number of satisfying assignments is odd or even. So, it is a weaker oracle than sharp P oracle because if you can count you can certainly determine the parity.

And now what we are saying is that but the reduction is not deterministic in the sense it direction is not full proof the correspondence allows some error. So, because of that we say it is BPP with access to parity SAT oracle.

(Refer Slide Time: 14:04)



In fact, we have seen something similar.

Theorem (VV): There is a randomized algorithm A ^{poly time} that takes as input a Boolean formula ϕ such that

$$\phi \in \text{SAT} \Rightarrow \Pr[A(\phi) \in \text{SAT}] \geq \frac{1}{2^n}$$

$$\phi \notin \text{SAT} \Rightarrow \Pr[A(\phi) \notin \text{SAT}] = 1.$$



So, what we are saying here is that polynomial hierarchy is contained in BPP with a parity SAT oracle. And what was our target? Our target was to show that polynomial hierarchy is contained in P to the sharp P oracle. So, P is a weaker class than BPP seemingly and sharp P is a stronger class than parity P. So, we want to come to the deterministic polynomial time but we are okay to use a counting oracle rather than a parity oracle.

So, this is what this is where we want to go to so ultimately, we will prove the statement in the right-hand side which is polynomial hierarchy is contain P to the sharp P but in this lecture, we will show the first statement. So, in fact like the this reduction that I just said when psi is true implies A psi is in a sense of parity SAT we have seen as somewhat of a similar statement in one of the previous lectures.

So, the statement being Valiant Vazirani theorem. So, what we showed there was given a Boolean formula phi. We have a randomized algorithm to construct another Boolean formula such that if phi is satisfiable then the other Boolean formula has exactly one satisfying assignment with high probability. So, it is a unique SAT instance, unique SAT is satisfiable.

(Refer Slide Time: 15:52)



that takes as input a Boolean formula ϕ
 such that $\phi \rightarrow \phi \wedge (h(x)=1)$
 $\phi \in \text{SAT} \Rightarrow P_n [A(\phi) \in \text{USAT}_{1/n}] \geq \frac{1}{8n}$
 $\phi \notin \text{SAT} \Rightarrow P_n [A(\phi) \in \text{SAT}] = 1.$
 where n is the number of variables.
 But the probability of success given by
 $1/8n$ is too small.
 In fact, it is an open question to boost the



ϕ is satisfiable then the reduction gives a unique SAT instance with a certain probability. If ϕ is not satisfiable the reduction will give you a unsatisfiable formula. So, if you recall the reduction was something like you mapped ϕ to ϕ and some kind of hash function. So, basically its ϕ and something. So, whenever the ϕ was unsatisfiable the formula in the right-hand side was also certainly unsatisfied it is because it is ϕ and something.

And the right where n is the number of so the probability of success in the first part was $1/8n$. Second part, it is a one-sided error. And notice that if, ϕ are satisfiable then it has then a ϕ has exactly one satisfying assignment with a certain probability. So, a ϕ having one satisfying assignment means it is a yes instance of parity SAT as well, one is an odd number. if ϕ is not satisfiable then a ϕ does not have satisfying assignment, so it is a no instance.

(Refer Slide Time: 17:06)



But the probability of success given by
 $1/8n$ is too small.
 In fact, it is an open question to boost the
 probability of success beyond $1/8n$.
 So we move to $\oplus \text{SAT}$.
 $A(\phi) \in \text{USAT}_{1/n} \Rightarrow A(\phi) \in \oplus \text{SAT}.$
 $A(\phi) \notin \text{SAT} \Rightarrow A(\phi) \notin \oplus \text{SAT}.$



So, we can actually view it like this if ϕ is satisfiable then this is sorry a ϕ is a yes instance of parity SAT. And if ϕ is not satisfiable then, ϕ is not in the parity SAT or rather it is a no instance of parity SAT, we can say it like this as well. Because 1 is an odd number and 0 is an even number. And so that is fine so with the Valiant Vazirani lemma theorem we are already somewhat close to here somewhat like what we have here.

What are the differences? The difference is that Valiant Vazirani lemma took one simple Boolean formula without any quantifiers. Here we are taking Boolean formula with k quantifiers everything is quantified there is no free variables. Second thing is that the probability of success is one-sided in Valiant Vazirani. So, the no instances are sure to go to no instances yes instances are going go to yes instances with a very small problem with a small probability $1/8^n$.

Whereas here in the target statement we want the correspondence to be much stronger the probability that if, i is true then a ϕ is in parity side is $1 - 1/2^m$. So, we need to do some kind of boosting and of course the third thing is that Valiant Vazirani deals with unique set so it is exactly one but whereas here we are okay with odd even thing. So, the reason we want to go to the odd event the parity from the unique SAT thing is that there is no clear way to boost the probability of success with the unique SAT setting.

So, in fact it is an open question to boost the probability of success beyond this $1/8^n$ even we are in the unique SAT setting. So, what we will do is to move to the parity set setting, which is automatic from the unique SAT because 1 means odd parity and 2 means even parity.

(Refer Slide Time: 19:23)

$A(\phi) \in \text{USAT}_{\text{yes}} \Rightarrow A(\phi) \in \oplus \text{SAT}.$
 $A(\phi) \notin \text{SAT} \Rightarrow A(\phi) \notin \oplus \text{SAT}.$

Valiant-Vazirani already gives us

$\Psi \text{ is true/sat} \Rightarrow \Pr_A [A(\Psi) \in \oplus \text{SAT}] \geq \frac{1}{kn}$

$\Psi \text{ is false/unsat} \Rightarrow \Pr_A [A(\Psi) \in \oplus \text{SAT}] = 0$

Suppose we have ϕ and Ψ . We can perform
 \wedge, \vee and \neg in the \oplus setting.



So, already the Valiant Vazirani give something like this if a formula psi is true meaning psi is satisfiable then A psi is so I will just say true or satisfiable false or unsatisfiable. The reason being when psi is true, I have to it has to be a simple Boolean formula. It cannot have quantifiers but then I use psi for a quantified fully quantified Boolean formula. So, when I say psi is true think of it as something like some there x x some phi of x or something.

So, there is some way to satisfy psi if that is the case then A psi has a maps to a plus instance of yes instance of parity SAT if it is false then A psi maps to a no instance of parity SAT or a sentence with 0 probability.

(Refer Slide Time: 20:27)

Suppose we have ϕ and Ψ . We can perform
 \wedge, \vee and \neg in the \oplus setting.

Let $\oplus_{\phi(x)}$ denote the parity of the count
 of x's that satisfy ϕ .

$\oplus_x (x_1 \vee x_2) = 1$

$\oplus_x (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) = 0$

Think of \oplus as a "quantifier" like $\exists x, \forall x$.



So, this is automatically this is what Valiant Vazirani, already gives us now we will see how to boost this probability of success. This is kind of what this valiant was running giving us

because the only issue is that this $\exists x \phi$ is not strictly speaking a Boolean formula. But we can view it as the first there exists you can take it as and fix the rest of the variables. Now suppose we have just like we performed arithmetic over the number of satisfying assignments.

We can also perform arithmetic over the parity of the satisfying assignments. Suppose we have formulas ϕ and ψ . Now we can create formulas depending on if I you can do a Boolean arithmetic with ϕ and ψ , let us see how. So, let us before that let us denote this symbol this notation as parity subscript $x \phi$ denote the parity of the count of the x as the satisfy ϕ . So, ϕ is a formula that has x as a free variable.

Then this indicates the parity $x \phi$ denotes the parity of the number of x as the satisfy ϕ . So, x could be a vector x need not be just one variable. So, just to give some examples so if, x is just two variables x_1, x_2 then parity of $x_1 \vee x_2 = 1$ because $x_1 \vee x_2$ has 3 satisfying assignments x_1 true x_2 false x_1 true x_2 true x_1 false x_2 true. At least one of them have to be true the non-satisfying assignment is when both of them are false.

Another formula is when x is $x_1 x_2$ consider this $x_1 \vee x_2$ and x_1 complement or x_2 complement. This has only 2 satisfying assignments 0 1 and 1 0 0 0 and 1 1 are not satisfying assignments so the parity is 0. This is just to give you an illustration of what this the symbol the parity x symbol is. So, you can think of it as a quantifier, just like there x is x something or for all x something you can think of parity access also as also a quantifier.

Parity x is just saying considering all x . So, there x is x is asking out of all the x is there at least one x that satisfies. For all x is saying are all the x is satisfying parity x is saying is exactly an odd number of x is satisfying. So, you can think of it as a quantifier as well.

(Refer Slide Time: 23:49)

Think of \oplus as a quantifier like $\exists x, \forall x$.


$$(\oplus_x \phi(x)) \wedge (\oplus_y \psi(y)) = \oplus_{x,y} (\phi(x) \wedge \psi(y))$$

Do the transformation that adds exactly 1 more sat assignment.

$$\neg (\oplus_x \phi(x)) = \oplus_x (\phi+1)(x)$$

$$(\oplus_x \phi(x)) \vee (\oplus_y \psi(y)) = \oplus_{x,y} (\underbrace{(\phi+1)(x)} \wedge \underbrace{(\psi+1)(y)}) + 1$$

Given Ψ , the VV reduction produces a randomly generated formula τ . We can repeat



So, now let us let us come to the arithmetic using these quantifiers. So, suppose you are given so basically, we want to say that we can do Boolean arithmetic using this you and still remain still give a parity instance. So, consider parity phi and parity psi and you want to take AND of them and it will still be a parity because you can take the multi you can take the product or the AND phi x AND psi y.

And as already mentioned this, the number of satisfying assignments of psi x sorry phi x and psi y this will be the product. So, if this has m and this has m prime this will have m times m prime satisfying assignments. So, the parity the AND says both of them are of positive parity meaning both of them have odd parity. So, but then that is true when m and m prime are m, m prime is odd if and only both of them both of m and m prime are all.

So, that is what it is if you want to negate the parity of psi x or phi x you can just add 1 to that we already saw how to manipulate the formula Boolean formula such that you can add something to get one more satisfying assignments or ten more satisfying assignments. So, if you add one more then the parity flips because odd becomes even with addition of 1 and even becomes odd.

So, negation can be written as just adding 1 when I add 1 I do not mean to say that you add 1 to the formula what I mean here is add or do the transformation that adds exactly one more satisfying assignment, that is what I mean by this. And finally, OR if you want to do OR of this if you have 2 formulas and you want to do OR the easiest so, we already saw negation and the easiest thing to do is de morgens.

So, you can do a negation and then again, a negation. So, negation is just adding 1. So, we do $5 + 1$ AND $\psi + 1$ and then you take in you add 1. So, basically, it is negation and then again indication. So, this is OR using de morgens, so basically what we are saying is that using parity we can perform the Boolean operations and still it will be a parity of some Boolean formula.

So, I can write the parity as I can write I can do logical Boolean operations. And still, it will be and get yet another parity Boolean operation Boolean formula.

(Refer Slide Time: 27:24)

The slide contains handwritten notes and a diagram. At the top right is the NPTEL logo. The notes describe the Valiant Vazirani reduction:

- Boolean formula:** $(\bigoplus_x \psi(x)) \vee (\bigoplus_y \dots) \dots$
- Text:** "Given ψ , the VV reduction produces a randomly generated formula τ . We can repeat the VV reduction many times to get $\tau^{(1)}, \tau^{(2)}, \dots$ "
- Diagram:** A circle labeled "all unique" with an arrow pointing to a smaller circle labeled "SAT instance". Below it, an arrow points to a box labeled "set assignment" with the mapping $\phi \rightarrow \phi \wedge (h(x)=0)$.
- Equation:**
$$VV \text{ reduction } A(\psi) = \psi \wedge \tau$$
- Equation:**
$$\text{Consider } \bigoplus_{i=1}^k \bigoplus (\psi \wedge \tau^{(i)}) = \bigoplus \left[(\psi \wedge \tau^{(1)} + 1) \dots \dots (\psi \wedge \tau^{(k)} + 1) \right] + 1$$
- Equation:**
$$= \bigoplus_{z} \Gamma(z)$$
- Text:** "The number of satisfying assignments... $n(\psi \wedge \tau)$ "

So, all these building blocks will be useful in the remaining steps. So, just to just to summarize what we have is the Valiant Vazirani theorem and this was our target. Given a sigma k instance sigma case at instance we want to get to a Boolean formula such that there is a high correlation. If the instance the sigma k instance is true, then the reduced instance should be a sense of parity side with high probability.

And the other way as well and this is a randomized reduction. So, now as I said there were the one of the things that we needed to do was to boost the probability of success in the valiant vazirani and I promise that I said that by moving from unique side to parity set we can boost the probability of success. So, let us see how we can do that? So, if you recall the valiant vazirani reduction, what did we do?

We had a Boolean formula and it may have it had it could have had many satisfying assignments. Let us say this is satisfying assignments and this is a set of all assignments. But

what did we wanted to transform to a formula which has exactly one satisfying assignment with high probability or with some probability. So, we can we what we did was we took AND with some so we took a hash function.

And we took AND with that hash function maps to 0. So, 0^m , so sum ϕ became ϕ AND $h(x) = 0$. So, $h(x) = 0$ is I am just somewhat abusing notation but I am just saying that this is how we transformed it. So, the point here is that this h of x was just chosen randomly so first if you remember we just chose m from 2 3 up to $n + 1$ and then for the range of number of satisfying assignments of ϕ .

And then so that what then that was done randomly. And then for that m we chose a hash function from a hash function pairwise uniform hash family. So, this was completely independent of this ϕ . So, it does not look at what ϕ is it just looks at ϕ to get the number of variables. So, this is what I am calling τ here, this is the same thing that I am calling τ here. So, what we can do is to repeat these many times.

So, we may get τ_1 τ_2 and so on, and the variant wizard and reduction was ψ and τ . So, now for this purpose for this part think of ψ as a Boolean formula, given ψ . So, later in the main theorem in this lecture we will think of ψ as a quantified Boolean form now you think of it as just Boolean formula. Now we want, so now we can repeat and boost so if ψ is satisfiable then with some probability it gets mapped to unique satisfying assignment.

So, how do we boost this one-sided errors? So, if you repeat it and you check for any of the instances does it have at least one SAT or exactly one satisfying assignment. And if any of the instances has exactly one satisfying assignment then ψ also had exactly one satisfying assignment because we know it is one sided if, ψ had no satisfying assignment then all τ_1 and τ_2 everything none of them will have any satisfying assignment.

Because ψ itself does not have. So, it is one sided error so this is how you do it. So, basically you are doing OR of any of this ψ and τ_1 ψ and τ_2 does any of them have exactly 1 satisfying assignment. And in order to boost it this exactly 1 does not scale. So, we will do it we will consider the parity so exactly once mean odd parity and no satisfying assignments mean even parity.

So, you want to take an OR of all this psi and tau i disparities the formulas. If any of them has a satisfying assignment then has an odd parity satisfying assignment then psi was a yes instance, so psi was also satisfiable this is what we know. And we already know how to do this? We already saw how to take OR of 2 formulas 2 parity formulas phi and psi so same thing we will do here.

We take psi and tau 1 + 2 psi n tau 3 + 2 and so on, psi n tau r + 1 which is r is the number of times we repeat this and finally you do plus 1 and take the parity of all and this entire thing you can convert it into 1 formula. The parity sign will be outside and it will be just formula. Let us say the formula the variable in the formula is z and your z is the collection of all the variables and you are taking parity over z.

So, now we can this repeated tau 1 you and this ANDs you get this one formula and psi is true or psi is satisfiable if and only if this gamma formula gamma had a has an odd number of satisfying assignments. So, not; if and only with a certain probability so now we need to compute the probability.

(Refer Slide Time: 34:07)

Consider $\bigvee_{i=1}^k \bigoplus (\psi \wedge \tau^{(i)}) = \bigoplus [(\psi \wedge \tau^{(1)} + 1) \dots \dots \dots (\psi \wedge \tau^{(k)} + 1) + 1]$

$= \bigoplus \Gamma(z)$

The number of random bits necessary = $O(m \ln)$

$P(\text{error}) \leq \left(1 - \frac{1}{2^m}\right)^{cm}$
 $\approx e^{-cm}$
 $= \frac{1}{2^m}$

$\psi(x) \in \text{SAT} \Rightarrow P_x [\Gamma \in \bigoplus \text{SAT}] \geq 1 - \frac{1}{2^m}$

$\psi(x) \notin \text{SAT} \Rightarrow P_x [\Gamma \in \bigoplus \text{SAT}] = 0$

How can we move to Σ_k -SAT?

So, I will come to the random bits soon. So, what we can do is, so the probability of success, what is the probability of success? The probability of maybe I will just write it here probability of success is probability of error may be easier to calculate it is actually, if psi is not satisfiable there is no error, we will always say no. What is the probability that when psi is satisfiable there is an error?

So, it is satisfiable but every instance we try we get a yes instance is mapped to no. So, this is at most so the probability of error in a single trial is $1 - 1/2^n$ and now let us say we raise it by some constant times c sorry some constant times m/n . So, this will roughly become some e^{-c} so e^{-c} power some constant times m and if you choose the constant carefully this there will be negative sign you will get $1/2^m$.

So, because its single sided error by repeated trials you get that the probability of error can be reduced to $1/2^m$. And the number where n is the number of variables, m is the m depends the target probability that you are seeking and c is some constant it will not be much, it will not it will not be dependent on m or n or anything. So, what we have now is that if ψ is satisfiable then γ is in the parity SAT with a higher probability with a high probability.

If ψ is not satisfiable then γ is in parity SAT with 0 probability. So, again it is one sided error but the error probability has become very small the property of success has become very high so from $1 - 1/2^m$, so we that is very close to 1. We started with one divided by 2^n . So, this shows that if ψ is a so what we have actually shown is that if ψ is a simple Boolean formula.

So, the first level of polynomial hierarchy let us say ψ is a Boolean formula so whether satisfiable or not is like an NP question, NP complete problem. That problem we are saying that we can reduce to checking deciding parity SAT and that will do with high probability. And the error is even 1 sided here and this looks very much like what we wanted to show this is what we wanted to show.

If ψ is true then instead of ψ is true now, we showed size satisfiable and the probability of success was $1 - 1/2^m$ in the case of yes instance and no instance it was actually 0. So, it was better than the no the case here but the only difference is that we showed the case when ψ is a formula with a single quantifier there exists x now, we want k quantifiers we want to extend to k quantifier so how do we do that.

(Refer Slide Time: 37:35)

$\Psi(x) \in \text{SAT} \Rightarrow \Pr[\Gamma \in \text{SAT}] = 0.$
 How can we move to $\Sigma_k\text{-SAT}$?
 We use induction + an oblivious version of VV theorem. Note that the \mathcal{C} 's produced by VV do not depend on Ψ except on the no. of variables.
 Theorem (VV-Oblivious): There is a poly-time randomized reduction A , that on input 1^n , outputs a Boolean formula $\mathcal{C}(x,y)$ where $x = (x_1, \dots, x_n)$ and y is a new set of variables such that for all formula β on n variables.



So, how can we move to k quantifiers? So, what we do is we just basically, it is what we saw already Valiant Vazirani and then we will do induction on top of that. So, it is just a bit more formal and I will be happy if you can understand what we have said so far. So, if you if you are trying to prioritize which part I want to follow and which part I will learn later then try to understand whatever we have seen so far. So, the next the next part gets a bit technical.

(Refer Slide Time: 38:17)

Theorem (VV-Oblivious): There is a poly-time randomized reduction A , that on input 1^n , outputs a Boolean formula $\mathcal{C}(x,y)$ where $x = (x_1, \dots, x_n)$ and y is a new set of variables such that for all formula β on n variables.
 $\exists x \beta(x) \Rightarrow \Pr_{x,y} [\beta(x) \wedge \mathcal{C}(x,y)] \geq \frac{1}{8^n}$
 $\nexists x \beta(x) \Rightarrow \Pr_{x,y} [\beta(x) \wedge \mathcal{C}(x,y)] = 0.$
 Proof of Main theorem: We use given $\Sigma_k\text{-SAT}$



So, but it is more of what we have done already. So, now this is again the valiant vazirani theorem, but just stated in a bit different form. So, I already mentioned this already the other points here already. If there is a formula beta on n variables Valiant Vazirani says if beta is satisfiable then we will produce another formula probabilistically such that the other formula has an exactly 1 satisfying assignment with height with some probability.

And we notice that this beta, this formula is independent of beta except for the number of variables the only thing that it. So, it works for any beta over n variables over the same number of variables. So, that same thing that I am writing here in fact I already said this over here I said this over here. So, we can so basically there is a polynomial time randomized reduction a that 1 puts input one power n which is just to obtain the number of variables in beta.

It outputs a Boolean formula tau, that has x and y as a variable. So, y's are the additional auxiliary variables it creates, where x is in x is the variables used in beta and y is some new set of variables such that if beta has a satisfying assignment, then the probability that the new formula has beta and tau has a satisfying assignment or has a unique satisfying assignment is 1 by 8 n.

(Refer Slide Time: 40:12)

$\exists x \beta(x)$ where $n = (x_1, \dots, x_n)$ where β is a new set of variables such that for all formula β on n variables. \rightarrow Unique

$$\exists x \beta(x) \Rightarrow \Pr_{x,y} [\bigoplus (\beta(x) \wedge \tau(x,y))] \geq \frac{1}{8n}$$


$$\nexists x \beta(x) \Rightarrow \Pr_{x,y} [\bigoplus (\beta(x) \wedge \tau(x,y))] = 0.$$

Proof of Main theorem: We are given $\exists x$ -SAT instance Ψ . Suppose $\Psi = \exists x \phi(x)$ where $\phi(x)$ has $k-1$ quantifiers. By induction hypothesis, there is a randomized

And if not if it is not satisfiable, beta is not satisfiable then it the probability that it has a satisfying assignment is itself 0. So, here it is just stated in the parity form if it has a unique satisfying assignment then the parity is 1. If it has no satisfying assignment the parity is 0. So, in fact this is actually Valiant Vazarani theorem says it is unique here and here it is 0. So, but then unique means parity 1 and 0 means parity even.

So, it helps to view the Valiant Vazarani in this setting because we already used the trick but I am just stating it again because we will apply this idea of getting this tau again and again.


(Refer Slide Time: 41:01)



$$\exists x \beta(x) \Rightarrow \Pr_{x,y} \left[\bigoplus_{x,y} (B(x) \wedge C(x,y)) \right] = 0.$$

Proof of Main theorem: We are given Σ_k -SAT instance Ψ . Suppose $\Psi = \exists x \phi(x)$ where $\phi(x)$ has $k-1$ quantifiers.

By induction hypothesis, there is a randomized reduction that for each fixed x , maps $\phi(x)$ to $\beta(x) = \bigoplus_z \rho(x,z)$ such that $\Pr_x \left[\bigoplus_z \rho(x,z) \right] \geq 1 - \frac{1}{2^{m+1}}$



So, the main theorem we are given, so let us come to the proof of the main theorem. As I already mentioned the main theorem states that you can take a sigma k SAT instance and get A psi instance a parity SAT instance. Such that if the sigma k instance psi is true then the parity SAT instance is true with high probability and if the sigma k psi instance psi is false then the parity SAT instance A psi is false with high probability.

And we already saw this when $k = 1$, when $k = 1$, it is just satisfiability and we already saw how this boosting can be done. So, this boosting is we use the fact that it is a parity and we could not do this boosting otherwise. We do not know how to do this boosting if it was this unique SAT setting. So, now let us say it is a sigma k instance psi and the so we say polynomial hierarchy if it is a pi k instance.

You can take the negation or you can view the pi k instance as a sigma k + 1 instance. Because if you can do that if you can decide this you can decide the negation as well. So, suppose this is sigma k instance psi and suppose psi is there exists x phi x, where phi x itself has k - 1 quantifier. So, the first quantifier is x are there exists and the variables that are quantified by the first quantifier. We call them x, so phi x itself has other variables that that have that I am not specifying here.

(Refer Slide Time: 42:59)



By induction hypothesis, there is a randomized reduction that for each fixed x ,

maps $\phi(x)$ to $\beta(x) = \bigoplus_z \rho(x,z)$

such that $\Pr_A \left[\phi(x) \equiv \bigoplus_z \rho(x,z) \right] \geq 1 - \frac{1}{2^{m+1}}$

For fixed x , parity of no. of z that satisfy $\rho(x,z)$.

Now let us sum $V.V.$ -additions $O(mn)$ times, producing $\tau^{(1)}(x,y), \tau^{(2)}(x,y), \dots, \tau^{(k)}(x,y)$.

$$\mathcal{X} = \bigvee_{i=1}^k \left[\bigoplus_{x,y} \beta(x) \wedge \tau^{(i)}(x,y) \right]$$


Now by induction so again the base case is done base case is what we already showed before this. Now by induction hypothesis we are doing induction on k by induction hypothesis there is a randomized reduction that given k and m it maps to an equivalent parity SAT instance. So, given $k - 1$ for this $\phi(x)$ that has $k - 1$ quantifiers it constructs a parity SAT instance called parity SAT instance called $\rho(x, z)$ and where x is for any fixed x .

So, once you fix the x , $\phi(x)$ for a fixed x $\phi(x)$ just becomes a Boolean formula or a Boolean formula on the remaining variables. Because x is fixed it is a formula with $k - 1$ quantifiers on the remaining variables. So, now on the same values of x I could write $\rho(x, z)$. So, when for a fixed x that is fixed on the left-hand side the x will be fixed on the right-hand side. So, for this, such that with high probability $\phi(x)$ the truth value of $\phi(x)$ will be the same as truth value of parity $\rho(x, z)$.

So, what I am saying is that given $\phi(x)$ so think of $\phi(x)$ is just a $k - 1$ quantified Boolean formula. We are getting $\rho(x, z)$ where x is fixed and z are the remaining variables. So, if you add up z or add up the number of ways to satisfy $\rho(x, z)$, upon so x is fixed so the things that can vary are just z . So, if you count the number of z for which $\rho(x, z)$ is satisfied. Then there is a $\phi(x)$ is true if and only if $\rho(x, z)$ has an odd parity with high probability.

Maybe I will just explain once again $\rho(x, z)$ means for fixed x parity of number of z that satisfy $\rho(x, z)$. A parity of number of z let me just write it again that satisfy $\rho(x, z)$ for a fixed x it is the number of the parity of this. Again, this is just the induction hypothesis. It is

the same statement, as a theorem but applied to a formula with $k - 1$ quantifiers ϕ and because ϕ has some free variables x . So, let us fix the x to something and then then say this.

(Refer Slide Time: 46:45)

But satisfy $P(x, z)$.

Now let us run VV-oblivious $O(mn)$ times,
producing $\tau^{(1)}(x, y), \tau^{(2)}(x, y), \dots, \tau^{(k)}(x, y)$.

$$K = \bigvee_{j=1}^k \left[\bigoplus_{x, y} \left(\underbrace{\bigoplus_{x, y} P(x, y)}_{\beta(x)} \wedge \tau^{(j)}(x, y) \right) \right]$$

For each j, x

$\beta(x)$ true $\Rightarrow P_n \left[\bigoplus_{x, y} \left(\beta(x) \wedge \tau^{(j)}(x, y) \right) \right] \geq \frac{1}{8n}$

$\beta(x)$ false $\Rightarrow P_n \left[\bigoplus_{x, y} \left(\beta(x) \wedge \tau^{(j)}(x, y) \right) \right] = 0$.

Now it is what we did already here what we what did we do above here we just took an OR with many tau's we can do the same thing again. We can run the valiant vazirani oblivious, that we already stated about order $m n$ times and we each time we get different tau values tau x , so tau 1 tau 2 and so on. And so let me denote this, the parity value of rho x, z by beta x . So, where the beta x denotes the truth value of parity of this it is just a Boolean formula.

So, what I will do is to, so maybe I just may be easier so just I just replace here that may be easier to view, think of it as a shorthand not as a separate formula. So, beta x I will actually I will use another colour to make it stand out, parity z rho x is it this is what it is. And this I am calling it as beta x . So, basically again I can with high probability ϕ x being true corresponds to beta x being yes instance of parity SAT.

So, think of beta as just a shorthand way of writing this. Now we just do the standard thing, we take n instances r instances of valiant vazirani the tau and check whether and when we know that valiant vazirani has a one-sided error and we can boost the probability of success. So, we consider the formula alpha which is the OR of these multiple instantiations of beta and tau z . And we know that for each tau we have this one-sided error probability.

That if beta is true then we have this there is some probability of it being a yes instance of parity SAT and if beta is false, it is not going to be a sense as a parity SAT.

(Refer Slide Time: 49:43)

Handwritten notes on lined paper showing a mathematical derivation. At the top right is the NPTEL logo. The main text consists of several lines of equations and annotations:

- Top line: $\Psi = \exists x \phi(x) \Rightarrow P_n [\exists x B(x)] \geq 1 - \frac{1}{2^{m+1}}$
- Second line: $\Rightarrow P_n [x \in \Theta \text{ SAT}] \geq 1 - \left(1 - \frac{1}{2^n}\right)^R - \frac{1}{2^{m+1}}$. A note "Choose R such that" points to the term $\left(1 - \frac{1}{2^n}\right)^R$, which is then equated to $\frac{1}{2^{m+1}}$.
- Third line: $\geq 1 - \frac{1}{2^m}$
- Fourth line: "Choosing $R = O(mn)$ such that $\left(1 - \frac{1}{2^n}\right)^R \leq \frac{1}{2^{m+1}}$ "
- Fifth line: $\exists x \phi(x) \Rightarrow P_n [\exists x B(x)] \leq \frac{1}{2^{m+1}}$

In the bottom right corner, there is a small video inset showing a person with glasses and a beard, wearing a green shirt, speaking.

So, now let us consider ψ which was the original formula, from Σ_k SAT. Now we assume that ψ is true there exists x such that $\phi(x)$ and we already know that by induction hypothesis. There is a correspondence between the truth correctness of $\phi(x)$ the satisfiability of $\phi(x)$ and $B(x)$ being a sentence of parity SAT or $B(x)$ being a sentence of parity SAT. So, the probability of that being $1 - \frac{1}{2^{m+1}}$.

So, now given β is yes instance of parity SAT now let us see how α performs. So, now what is the probability that α is a yes instance of parity SAT. So, remember α is this OR of these multiple instantiations of β . So, now what is the probability that α is a yes instance of parity SAT? It is $1 - \frac{1}{2^{m+1}}$. And what is the probability that β is? It is a no instance.

It is a problem so remember the assumption is that β is a yes instance. And so, what is the probability that it is α is a no instance? $1 - \frac{1}{2^{m+1}}$ each of these r times that we instantiate τ , all of them lead to a no instance. So, the probability of error is $\frac{1}{2^{m+1}}$ the probability of success of $1 - \frac{1}{2^{m+1}}$ such trial is $1 - \frac{1}{2^{m+1}}$, and the probability of error is $\frac{1}{2^{m+1}}$. And the probability all of the r trials or failures is $\left(\frac{1}{2^{m+1}}\right)^r$ or the term that is over here.

And from this over here, and so that is the probability that β for the correct β α turns out to be incorrect and then we have also this term which is $\frac{1}{2^{m+1}}$ term which is coming from the β itself the probability of β itself being incorrect. And

we can choose r such that the first term this term we can choose r such that this term is also 1 divided by $2^{\text{power } n + 1}$.

We can choose r I have written in the bottom r to be order m, n such that $1 - 1$ divided by $8, 10$ whole power r is 1 divided by $2^{\text{power } n + 1}$. So, this can be done and so $1 - 2^{\text{power } n + 1}$ sorry 1 divided by $2^{\text{power } n + 1}$ and again 1 divided by $2^{\text{power } n + 1}$ you add that up to get 1 divided by $2^{\text{power } m}$. So, the whole probability that ψ is a true instance yes instance and yet the α is no instance is 1 divided by $2^{\text{power } n}$. So, that is the probability of success when ψ is true.

(Refer Slide Time: 53:05)

$- 2^{m+1}$

$\geq 1 - \frac{1}{2^m}$

Choosing $R = O(mn)$ such that $(1 - \frac{1}{2^n})^R \leq \frac{1}{2^{m+1}}$

$\Pr(\phi(x)) \Rightarrow \Pr[\exists x B(x)] \leq \frac{1}{2^{m+1}}$

$\Rightarrow \Pr[\alpha \in \text{SAT}] \leq \frac{1}{2^{m+1}} < \frac{1}{2^m}$

We finally note that α can be written as

$\bigoplus_z Q(z)$ for some formula $Q(z)$.

When ψ is false which means there is the does not exist in x for which ϕ is satisfied. Now what is the probability that β is the correct β ? Again, this is coming from the previous level of induction so that the probability of error is 1 divided by $2^{\text{power } n + 1}$. Now suppose so which means β is a no instance the probability that β is a yes instance is 1 divided by $2^{\text{power } n + 1}$. And what is the probability that α is a yes instance?

So, we know that if α, β is a no instance, then all the α will always be no instance because you are taking no instance and add of that it will always be a no instance. The error in this process is one-sided. So, the only error can happen only when β itself to begin with was a yes instance. So, recall we are talking about no instance of ψ . And what is the probability that β was in yes instance? 1 divided by $2^{\text{power } m + 1}$.

And that is the only probability of error that can occur. So, $1 - 2^{-n+1}$ is at most $1 - 2^{-m}$. So, in either case whether ψ is a yes instance or no instance the probability that α is consistent with ψ is at least $1 - 2^{-n}$. So, that is how we get these probabilities and that proves the probability that we wanted in the main theorem. If ψ is true the probability that.

So, the α here will be α is a yes instance of parity SAT is $1 - 2^{-m}$ and ψ is false ψ is false means the probability that yes instance $1 - 2^{-n}$.

(Refer Slide Time: 54:52)

We finally note that α can be written as

$$\bigoplus_z \bigoplus_{j=1}^k \left[\bigoplus_{x,y} \left(\bigoplus_{z} \theta(x,y,z) \wedge \tau_j(x,y) \right) \right]$$

We have already seen how to take the \bigvee inside the \bigoplus . Finally, we note that $\bigoplus \circ \bigoplus$ is simply \bigoplus . For example.

Now the final small thing that I wanted to mention is that α is this formula. It is an OR of many parity SAT instances AND with τ_j . So, the only thing to note is that α itself is a parity SAT instance. So, this is not really that difficult to see because we have already noted that we can operate we can do all these operations we can do AND, OR negation everything can be done in the parity world itself.

So, it is just that is remaining just one second, so α is this and we can write α is OR of this parity of $\theta(x, z)$ and τ_j and maybe it see simpler to write this entire thing the entire highlighted thing as parity of $\theta(x, y, z)$ $\theta_j(x, y, z)$. And we have already seen how to take the OR inside the parity, basically we use the de Morgan's law. So, we added one then had the product and then added one again.

(Refer Slide Time: 56:06)

the \oplus . Finally, we note that \oplus of \oplus is simply \oplus . For example.

$$\alpha = \bigoplus_{x,y,z} \left[\left(\bigoplus_{x,y,z} \oplus^{(1)}(x,y,z) + 1 \right) * \left(\bigoplus_{x,y,z} \oplus^{(2)}(x,y,z) + 1 \right) * \dots \right] + 1$$

$$= \bigoplus_{x,y,z,w} \left[\dots \right]$$

So, we can do the same operation again. So, we can have a new parity sign outside maybe call it w just to take care of additional variables if any and then priority of theta 1 x, y, z + 1 parity of theta 2 x, y, z + 1 and so on and finally we add another + 1 outside. So, we will get another big Boolean formula and we can take all the parities inside because what one can notice that parity of parity is also parity.

So, if you sum up some items sum up some integers and if you know the sum is odd then you know that exactly odd number of the inputs must be odd. Odd number of the terms must be odd if you have an even number of terms that is all then the sum will be even. So, you can merge these parities to get 1 big parity and you can represent alpha as one big parity formula. So, that completes the proof.

So, the main ingredients of this proof again what we want what we have shown is that polynomial hierarchy is in BPP to the parity P oracle BPP with the parity P oracle. And we wanted to show that given a sigma such instance we can reduce it in a randomized fashion to a parity SAT instance. Such that if the input is a yes and since the output is a yes instance with high probability.

So, the main ingredient was repeatedly the fact that valiant vazirani was an oblivious reduction that the input we just took an AND with some Boolean formula. And the Boolean formula was just completely oblivious of what the original formula was except for the fact that it looked at the number of variables. And then we noted that using parity operation. We

can do all this operation again we can represent AND we can represent OR all of this can be accomplished using parity.

And therefore, this allows us to boost the probability of success. So, starting from the Valiant Vazirani, we boosted the probability of success to get the first base case of the induction which was over here this is what we had. And then for the remaining Σ_k SAT we use induction. And assuming the previous levels formula by induction hypothesis if that is a parity SAT formula already.

We could we could again use the same trick, we could again use the Valiant Vazirani sensation to get sorry again to get a new formula that is a parity, parity SAT formula. And we can choose m such that the probability of success is high enough and in all of this the number of repeated trials that we required is always polynomial. So, in this case it is order m^n which is polynomial and we have at most k levels of k levels in the polynomial hierarchy.

So, it is all related polynomial in the length of the input formula. And so, what is left? What we have shown is that polynomial hierarchy is in BPP with a parity P oracle. What we want to show is that it is in P with a deterministic polynomial time with a sharp P oracle and this we will show in the next lecture. So, we want a somewhat of a weaker class as the base class instead of BPP we do not want randomness.

We want this deterministic polynomial time but then we will trade the parity instead of the parity P oracle we will actually use a counting oracle and this is where the counting will matter. And with that I think I will stop this lecture. Thank you.