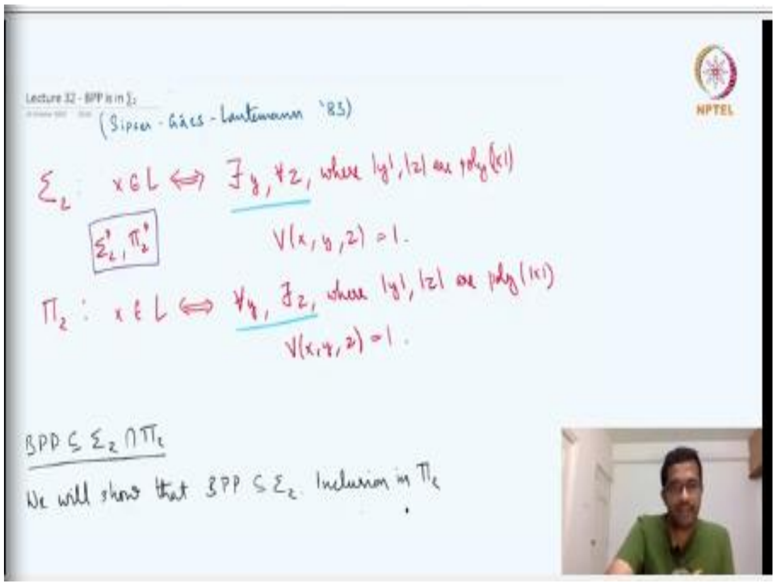
# Computation Complexity Theory

**Prof. Subrahmanyam Kalyanasundaram**
**Department of Computer Science and Engineering**
**Indian Institute of Technology-Hyderabad**

## Lecture-32
## BPP is in Polynomial Hierarchy

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



Lecture 32 - BPP is in $\Sigma_2$

(Sipser - Gács - Lautemann '83)

$\Sigma_2$: $x \in L \Leftrightarrow \underline{\exists y, \forall z,}$ where $|y|, |z|$ are poly$(|x|)$

$\boxed{\Sigma_2^p, \Pi_2^p}$ $V(x, y, z) = 1$.

$\Pi_2$: $x \in L \Leftrightarrow \underline{\forall y, \exists z,}$ where $|y|, |z|$ are poly$(|x|)$

$V(x, y, z) = 1$.

$\underline{BPP \subseteq \Sigma_2 \cap \Pi_2}$

We will show that $BPP \subseteq \Sigma_2$. Inclusion in $\Pi_2$

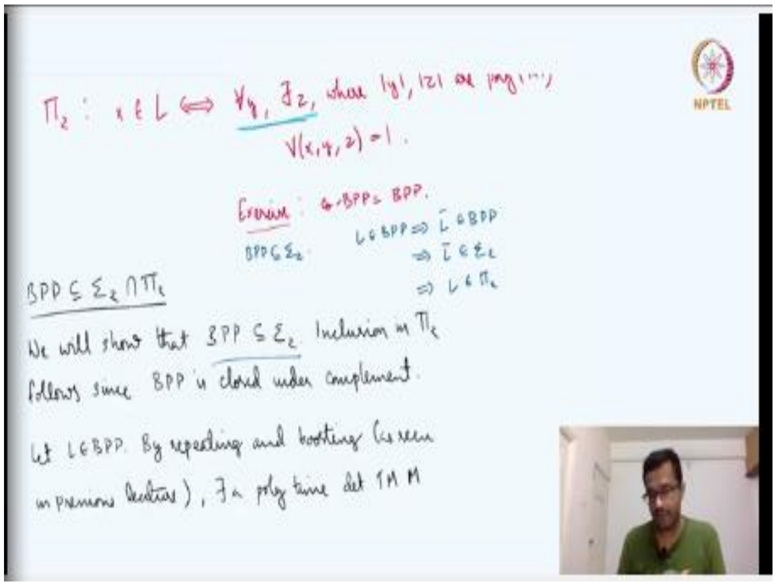
Hello and welcome to lecture 32 of the course computational complexity. So, we have been seeing randomized complexity classes such as RP, Co-RP, ZPP, BPP and so on. Before that we saw a polynomial hierarchy, this is we saw sigma 2 pi 2, sigma 3 pi 3 and so on, which were defined as follows. So, just to that you could recollect I have recalled the definitions here sigma 2 is a class of all languages X sorry l such that X is in l if and only if you can verify it using 2 quantifiers.

Such that X is in l if and only if that X is y such that for all z the verifier can verify that the existence of X using y and z. And pi 2 has 2 quantifiers also, but it appears in the other order first for all quantifier followed by their axis quantifier. So, and this is a sigma 2 and pi 2, this is the definition that we saw. Later we also saw another definition using oracle turing machines, we saw that sigma 2 is equivalent to NP with the cytorectal and pi 2 is equivalent to co-NP with the cytorectal, we saw that.

And one would think that the BPP the randomized complexity classes and these are kind of different beasts. So, on one hand you have quantifiers, alternations to do oracles and so on, and the other hand we have this randomness bit strings are generated randomly and that is used to verify. Interestingly, in this class we will show that BPP is actually contained in sigma 2. In the previous lecture I mentioned that it is not known whether BPP is contained in NP. However it was shown that BPP is even though we still do not know whether DP is in NP, BPP is in NP, now we know that it is in sigma 2.

And this was showed by Sipser-Gacs-Lautemann in 1983. And what happened is Sipser showed that BPP is contained in the polynomial hierarchy followed by Gacs who showed that it is in sigma 2, in fact sigma 2 and pi 2. So, in fact we will see that BPP is in sigma 2 as well as pi 2. And further Lautemann simplified the proof and made it more easier to present and in fact that is the version that we will see in today's lecture.

**(Refer Slide Time: 03:06)**



$\Pi_2$ : $x \in L \Leftrightarrow \underline{\forall y, \exists z,}$ where $|y|, |z|$ are poly(...)

$V(x,y,z) = 1$.

Exercise : co-BPP = BPP.

BPP $\subseteq \Sigma_2$

$L \in$ BPP $\Rightarrow \bar{L} \in$ BPP

$\Rightarrow \bar{L} \in \Sigma_2$

$\Rightarrow L \in \Pi_2$

$\underline{\text{BPP} \subseteq \Sigma_2 \cap \Pi_2}$

We will show that $\underline{\text{BPP} \subseteq \Sigma_2}$. Inclusion in $\Pi_2$ follows since BPP is closed under complement.

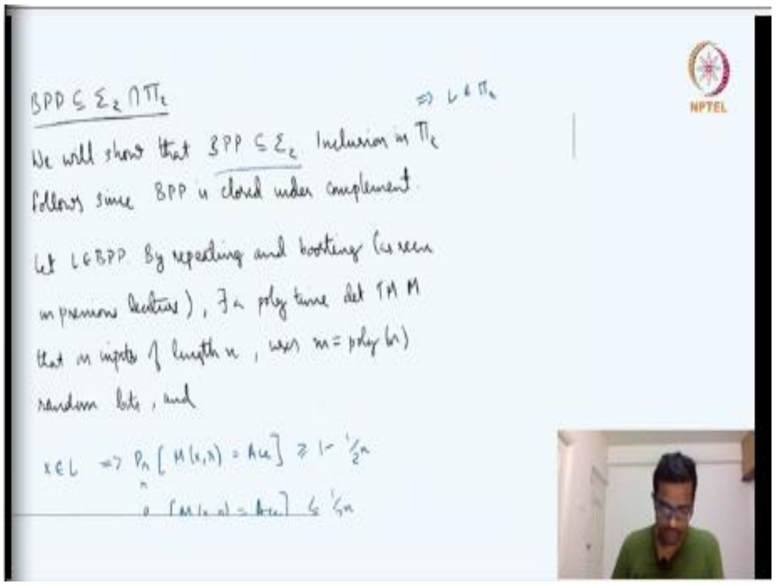
Let $L \in$ BPP. By repeating and boosting (as seen in previous lectures), $\exists$ a poly time det TM M

So, BPP is contained in sigma 2 intersection pi 2 that is the proof that we will see today. So, first up again it may not be difficult to see, if I have not shown this before I want you to verify this that BPP = co-BPP. So, again by now it should be clear what I mean by co-BPP, co-BPP is a class of all languages whose complement is in BPP. So, basically if you take it in other words BPP is closed under complement.

If you take a language in BPP take it is complement even the complement is in BPP that is what I am asking to show. And once you have that, and suppose we show BPP is in sigma 2, suppose we show this. Now this implies, if n is in BPP or rather I will remove the if and just write it as a implications, L is in BPP. Then L complement is also in BPP, this is from what we said BPP is closed under complement which implies that if BPP is contained in sigma 2 L complement is in sigma 2, which means that if you take a language in sigma 2 and take it is complement that should be in pi 2.

So, L is n pi 2, so L complement is in sigma 2 implies L is n pi 2. So, just by the 2 facts, BPP is contained in sigma 2 and the fact that BPP is closed under complement together we get that BPP is contained in pi 2 as well. So, what we will show is that BPP is in sigma 2, inclusion in pi 2 is just what I described here. So, again you can stare at what I have written in the top right part and try to follow, otherwise you could ask questions.

**(Refer Slide Time: 05:28)**



$\underline{BPP \subseteq \Sigma_2 \cap \Pi_2}$ $\Rightarrow L \in \Pi_2$

We will show that $\underline{BPP \subseteq \Sigma_2}$. Inclusion in $\Pi_2$ follows since BPP is closed under complement.

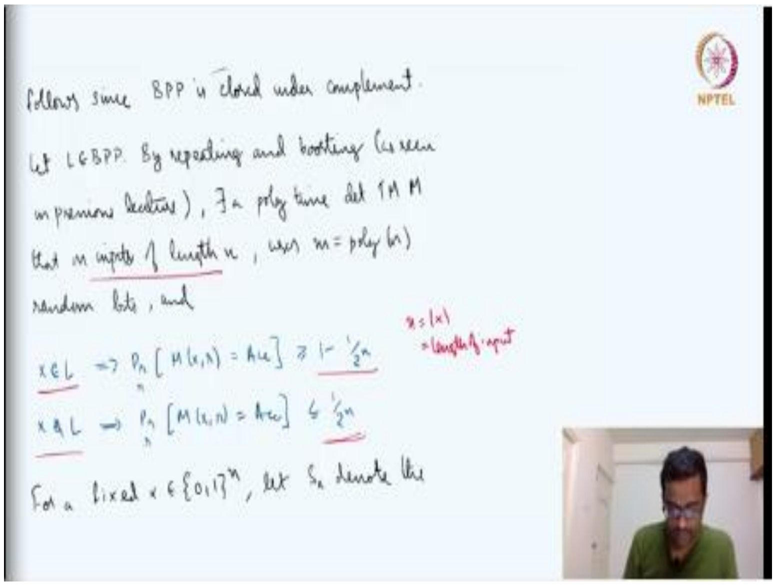
Let $L \in BPP$. By repeating and boosting (as seen in previous lectures), $\exists$ a poly time det TM M that on inputs of length n, uses $m = poly(n)$ random bits, and

$x \in L \Rightarrow \Pr_r[M(x,r) = Acc] \geq 1 - \frac{1}{2^n}$

$\Pr_r[M(x,r) = Acc] \leq \frac{1}{2^n}$

Now let us first see how what is a high level idea of showing BPP's in sigma 2?

**(Refer Slide Time: 05:39)**



Follows since BPP is closed under complement.

Let $L \in BPP$. By repeating and boosting (as seen in previous lecture), $\exists$ a poly time det TM M that on $\underline{\text{inputs of length } n}$, uses $m = poly(n)$ random bits, and

$\underline{x \in L} \Rightarrow \Pr_{r}[M(x,r) = Acc] \geq \underline{1 - \frac{1}{2^n}}$ $n = |x|$ = length of input

$\underline{x \notin L} \Rightarrow \Pr_{r}[M(x,r) = Acc] \leq \underline{\frac{1}{2^n}}$

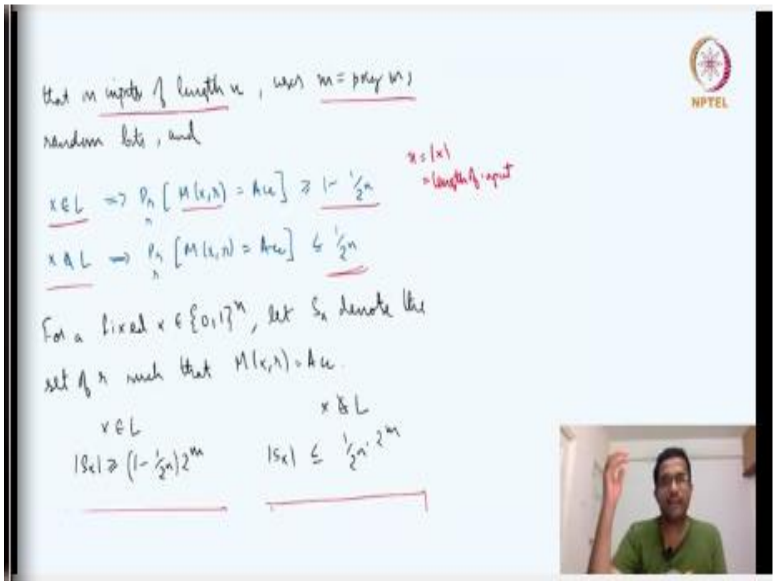
For a fixed $x \in \{0,1\}^n$, let $S_x$ denote the

First up we saw how we could boost the probability of success in BPP. We could do the same thing and so in fact what we saw in the previous lecture was that if we had a machine that accepts with probability half + 1 by polynomial in X for X in the language. And accepts with probability at most half - 1 by polynomial X for X not in the language.

Then you can boost it to get if X is in l probability of acceptance is at least two thirds and X is not in the property of acceptance is at most one thirds. So, we could get to two thirds, one thirds. So, now we could pretty much do the same proof or same idea but maybe the parameters need to be tweaked and get the following characterization. If X is in the language then we could basically by repeating or boosting the two thirds, one thirds machine we could boost up the probability of acceptance for X is an l to 1 - 1 by 2 power n.

And when X is not in l the probability of acceptance is at most 1 divided by 2 power n, what is n here? By n I denote the length of the input, so n here is the size of X or which is the length of the input, again I have written here also n is the length of the input.

**(Refer Slide Time: 07:22)**



that on inputs of length n, uses $m = \text{poly}(n)$ random bits, and

$n = |x|$
$=$ length of input

$x \in L \Rightarrow \Pr_r [M(x,r) = \text{Acc}] \geq 1 - \frac{1}{2^n}$

$x \notin L \Rightarrow \Pr_r [M(x,r) = \text{Acc}] \leq \frac{1}{2^n}$

For a fixed $x \in \{0,1\}^n$, let $S_x$ denote the set of $r$ such that $M(x,r) = \text{Acc}$.

$x \in L$ | $x \notin L$

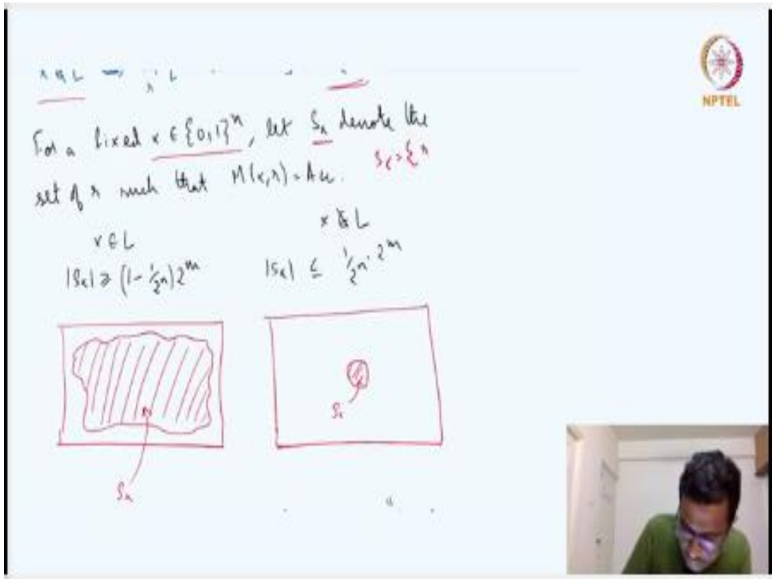
$|S_x| \geq (1 - \frac{1}{2^n})2^m$ | $|S_x| \leq \frac{1}{2^n} \cdot 2^m$

And again just as a reminder what is this probability taken over? Probability is taken over all the random choices, so once again probability is not over the input. For any input X even after you fix the input the probability can be taken over the random choices. And let us say, so for a fixed length input the random input is also fixed if the input X is fixed. If the input X is of length n then the length of the random bits are also fixed.

Let us say the random bits are of length M, where M is some polynomial in n, where n is the length of the input x. So, if x is the input and r is the random bit string, so x is of length n and r is of length M. Then we denote by M x, r we denote the randomized turing machine which takes x as input and r is a random bits. So, M performs in perfectly a completely deterministic manner but all the randomness comes from the fact that r is random, the r the random bit string is random.

So, M once r is fixed everything is deterministic, it is like saying in randomized quick sort. Once I tell you which what random numbers are going to appear then you know which pivots are going to be chosen, everything is deterministic from there on. So, the point here is that when x is an L you can boost the probability of acceptance to 1 - 1 by 2 power n, again n is the length of the input. And when x is not in L, we can reduce the probability of acceptance to 1 divided by 2 power n and this can be done by the same boosting argument just by tweaking some parameters.

**(Refer Slide Time: 09:18)**



For a fixed $x \in \{0,1\}^n$, let $S_x$ denote the set of $r$ such that $M(x,r) = Acc$. $S_x \subseteq \Sigma^n$

$x \in L$

$|S_x| \geq (1 - \frac{1}{2^n}) 2^m$

$x \notin L$

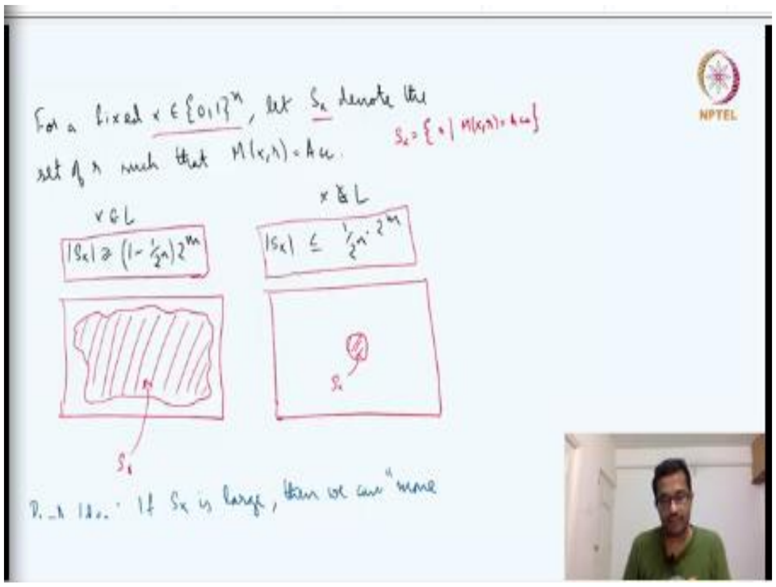
$|S_x| \leq \frac{1}{2^n} \cdot 2^m$

So, now let us see a different perspective of how this randomized algorithms work. Again this perspective also may be useful in understanding randomized algorithms separately. But this perspective is particularly helpful for this proof. So, given in string x, I had spoke about the probability of acceptance over a random set of random bits. So, let r, so I said r the random bit string is going to come from is going to have length M.

So, which means it is going to be there are 2 power M possible choices of r. So, assuming all of them are equally likely, when I say the probability of acceptance is half this means that or when I say probability of acceptance is 80% let us say. This means that out of this 2 power M possible random bit strings 80% leads to acceptance and 20% leads to reject. Similarly if I say that the probability of acceptance is only 10% this means out of this 2 power M random strings on the input x.

Once we fix the input x out of this 2 power M random strings 10 % leads to accept and 90% leads to reject. So, now let us call for a fixed x in 01 power n, let us call S x, what is S x? S x is a set of random strings for which x is accepted. So, more formally I will write here S x is a set of all r.

**(Refer Slide Time: 11:06)**


For a fixed $x \in \{0,1\}^n$, let $S_x$ denote the set of $r$ such that $M(x,r) = Acc$. $S_x = \{r \mid M(x,r) = Acc\}$

$x \in L$ $\boxed{|S_x| \geq (1 - \frac{1}{2^n})2^m}$

$x \notin L$ $\boxed{|S_x| \leq \frac{1}{2^n} \cdot 2^m}$

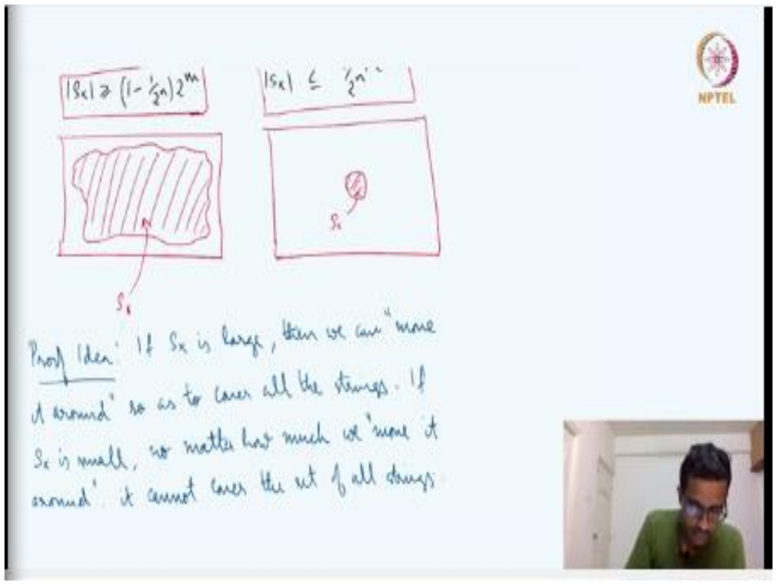
Main Idea: If $S_x$ is large, then we can "move

So, here the machine is fixed, x is fixed, is the set of all r such that M x r needs to accept. So, if x is in the language S x will be bigger if x is not in the language S x will be smaller, that is what we want. Now we have done this boosting, again this boosting when x is in the language the probability of acceptance is 1 - 1 divided by 2 power n. That means out of the 2 power n possible strings 2 power M possible random strings 1 - 1 divided by 2 power n fraction or this many random strings lead to acceptance.

So, S x is of size at least 1 - 1 divided by 2 power n whole multiplied by 2 power m, when x is in the language. When x is not in the language S x is of size at most 1 by 2 power n fraction which is 2 power m divided by 2 power n. So, let us try to see this pictorially which is what I have drawn below, when x is in the language let this box denote the universe of all the strings of length to m.

So, there are 2 power n such strings, so a huge majority of these strings lead to acceptance when x is in L. So, I will call this S x and I have drawn pictorially, so S x you can see covers almost the entire rectangle, when x is not in L you can see in the right hand side S x is much smaller, it is a much smaller part far from covering it, it is covering only a small portion, it is only reaching a small portion.

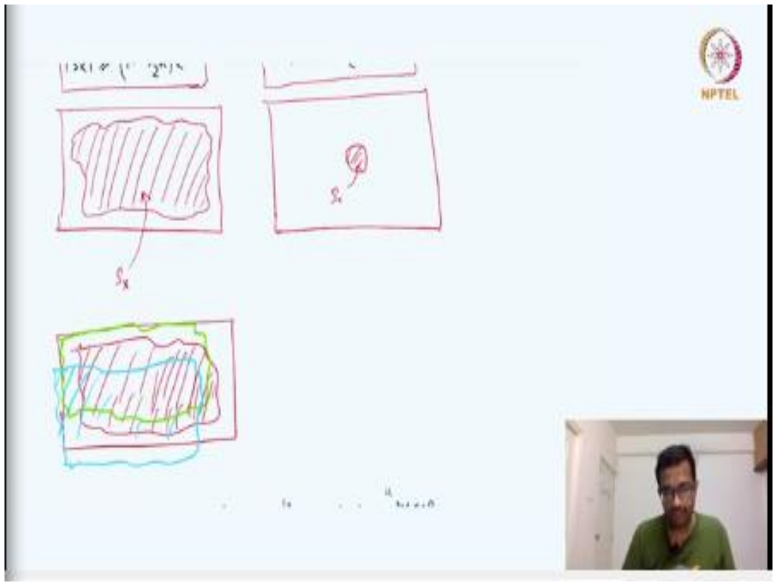
**(Refer Slide Time: 13:12)**


$|S_x| \geq (1 - \frac{1}{2n}) 2^m$
$|S_x| \leq \frac{1}{2n}$
$S_x$
$S_x$

Proof Idea: If $S_x$ is large, then we can "move it around" so as to cover all the strings. If $S_x$ is small, no matter how much we "move it around" it cannot cover the set of all strings.

So, with this pictorial idea in firmly in mind I can explain the high level proof. So, this is one of these proofs that have a very clean interesting visualization at the high level. Of course you can go to details delta epsilon calculations can be done for all the proofs but this proof has a very clean visualization of what is happening at the high level. And once if you understand this the rest is something that you could naturally work out on your own. So, when x is in the language S x is large as we see in the left hand side. So, I will just make some space, so that I can I can write this.

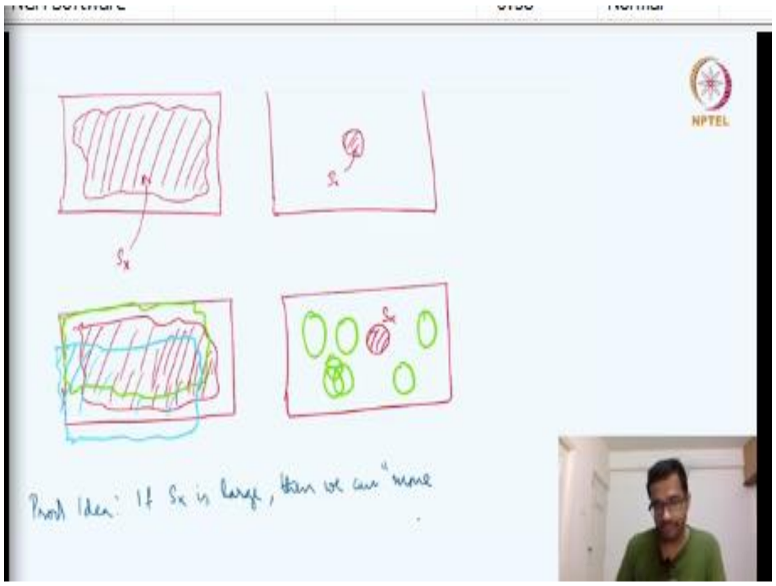
**(Refer Slide Time: 14:06)**



So, when x is in the language S x is large, so what happens when S x is large is again S subscript x. So, now consider the same thing the similar image here suppose this is S x consider now let us move around S x slightly, so now let me consider slight shifting of it, maybe I will shift it slightly, maybe I will denote it with a green colour something like this.

So, now it is covering some top part, now let us say I move in to the bottom some slightly, so maybe something like this. And you can see when I am shifting this, it is covering some regions of the original red rectangle that were not covered by S x. So, the blue part cover some part, green part to cover some other part that was missed etcetera. So, if we move around S x we can possibly cover the entire universe of the m length strings by this thing. So, as it is you can possibly see that by 5 or 6 shifts you may end up covering the entire thing.

**(Refer Slide Time: 15:56)**

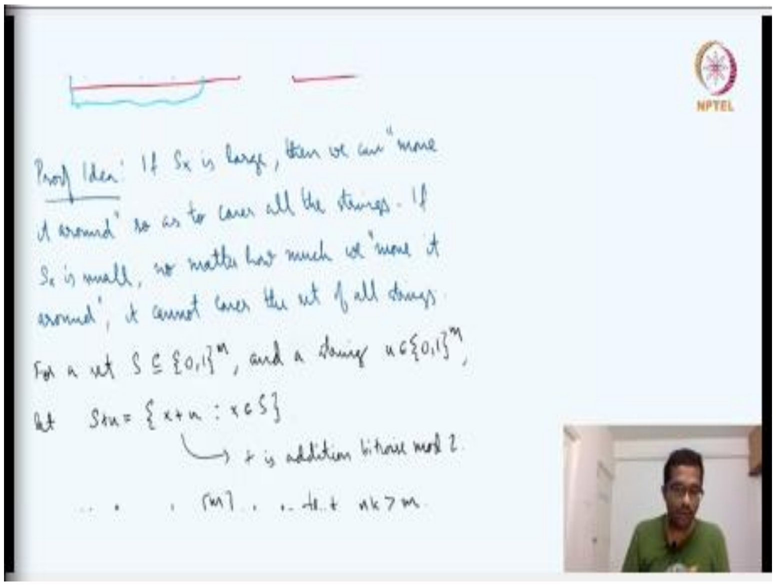


Proof Idea: If $S_x$ is large, then we can "move"

However when S x is small, S x is so small, now even if I have multiple copies of S x, so this is S x. Now even when I have multiple copies of S x, this is not going to cover the thing. So, in the case of in the left hand side we had even with S x is so huge, it covers so much of the area that even by shifting around you may end up covering by shifting around 2, 3 times or 4, 5 times you may end up covering the entire thing.

Let us say maybe 10 times you cover the entire thing. But here 10 times even if they are all disjoint even then they are not covering in the right hand side. And if there is an overlap between some of them then it covers even less, so even though there are 4 or 5 copies of S x but then they are all overlapping, so they cover even less.

**(Refer Slide Time: 16:59)**



<u>Proof Idea</u>: If $S_x$ is large, then we can "move it around" so as to cover all the strings. If $S_x$ is small, no matter how much we "move it around", it cannot cover the set of all strings.

For a set $S \subseteq \{0,1\}^m$, and a string $u \in \{0,1\}^m$,

let $S+u = \{x+u : x \in S\}$

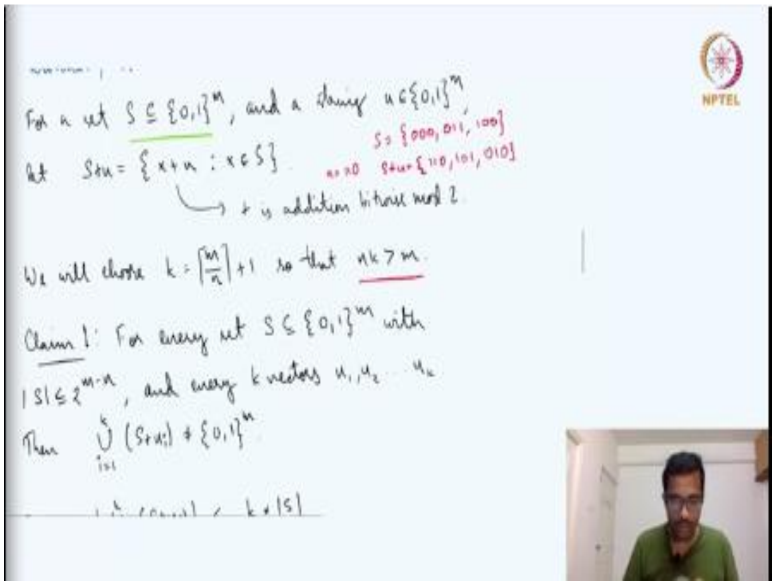
$\hookrightarrow$ $+$ is addition bitwise mod 2.

[illegible] $\lceil m \rceil$ [illegible] $nk > m$.

So, the proof idea is this. If S x is large then we could shift the S x around and all the strings in the universe can be covered as you can see using some small number of shifts. However when S x is small if you move it around and shift and if we limit the number of doing this operation moving and shifting, it is not going to be able to cover all the set of strings.

No matter like let us say if you limit to 10 movements and shifts, it is not going to cover this entire area. This is because the area is so big, so more big than even 10 times the area of S x, so even if you spread it all out even if you could do that it is not going to help. So, that is the main idea of the proof and that is going to lead us to a sigma 2 characterization. Again the goal remember is to show that BPP is in sigma 2. So, now let us try to see what I mean by shifting the S x? What do I mean by shifting a set?

**(Refer Slide Time: 18:12)**



For a set $S \subseteq \{0,1\}^m$, and a string $u \in \{0,1\}^m$,

let $S+u = \{x+u : x \in S\}$

$\hookrightarrow$ + is addition bitwise mod 2

$S = \{000, 011, 100\}$

$u = 110$ $S+u = \{110, 101, 010\}$

We will choose $k = \lceil \frac{m}{n} \rceil + 1$ so that $nk > m$.

Claim 1: For every set $S \subseteq \{0,1\}^m$ with $|S| \leq 2^{m-n}$, and every $k$ vectors $u_1, u_2 \ldots u_k$

Then $\bigcup_{i=1}^{k} (S+u_i) \neq \{0,1\}^m$

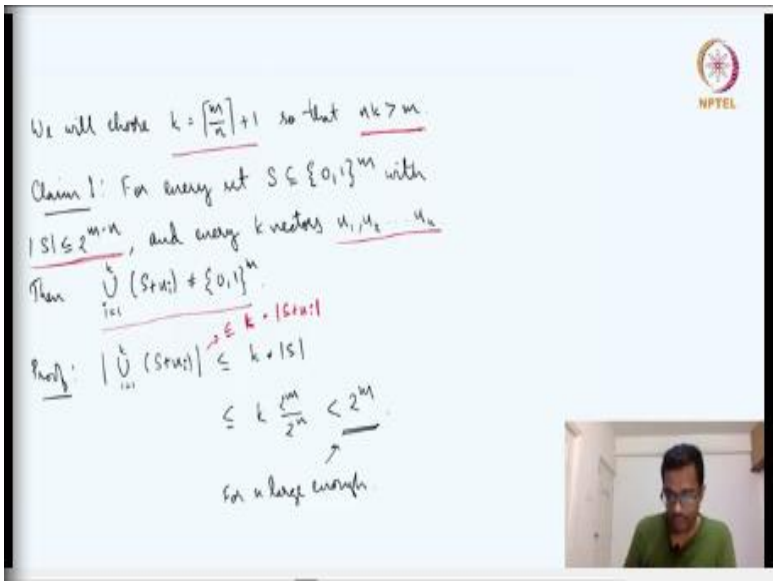
$\cdots \leq k \cdot |S|$

So, consider the set S x, also consider a set S this could be anything, S is a subset of 0 1 power m meaning set of all m length strings, it is a subset of m length strings. And consider a specific m length string called u, now by S + u I denote adding u to all the strings in S. So, maybe a very simple example in the right side, if let us say S is 000, 011 and 100 and let us say u is 110 then S + u is nothing but you add 110 to all the strings, so you add 110 to 000 you get 110, you add 110 to 011.

So, notice that this addition is bitwise mod 2 or rather bitwise x r, there is no carry or anything. So, 1 + 0 is 1, 1 + 1 is 0, 0 + 1 is 1 and 110 and 100, it is going to be 010 it is bitwise x r or bitwise mod 2 addition. So, this is the operation that I refer to by shifting around. So, this is the shifting operation and we will choose a number k again this will appear later to be m divided by n + 1 or with the ceiling, m divided by n may not be an integer.

So, by this what is assured is that n times k will always be bigger than m, so this is what we want n times k to be bigger than m is what we want, this is the smallest such k. We want k to be an integer such that n times k is bigger than n and smaller such integer is what we choose. So, I hope the high level idea is clear, the fact is that when x is in the language we can move around S x some number of times let us say 10 times and cover the entire universe. When x is not in the language, no matter how we move it around 10 times is not going to be enough to cover the entire universe.

**(Refer Slide Time: 20:48)**



We will choose $k = \lceil \frac{m}{n} \rceil + 1$ so that $nk > m$

Claim 1: For every set $S \subseteq \{0,1\}^m$ with $|S| \leq 2^{m-n}$, and every k vectors $u_1, u_2 \ldots u_k$

Then $\bigcup_{i=1}^{k} (S + u_i) \neq \{0,1\}^m$.

Proof: $\left| \bigcup_{i=1}^{k} (S + u_i) \right| \leq k \cdot |S|$ $\leq k \cdot |S + u_i|$

$\leq k \frac{2^m}{2^n} < 2^m$

For n large enough

And the k here is going to be our 10 or k is going to be the fixed number of shifts and repeats that we are allowed. And this will be used to derive a sigma 2 characterization. Let us first show the second part that when x is not in the language, no matter what shifts we select k shifts are not going to be enough and that is just going to be by counting the area of k shifts.

The first claim, so I am writing it as 2 claims for any set S with the size of S being less than 2 power m - n. So, notice that this is what we said here when x is not in l size of S x is at most 2 power m divided by 2 power n, this is what we have in the red box here when S x, x is not in l, it is this is what I am referring to. When S here in the claim 1 when S is such a set S, so which is satisfied by S x when x is not in l.

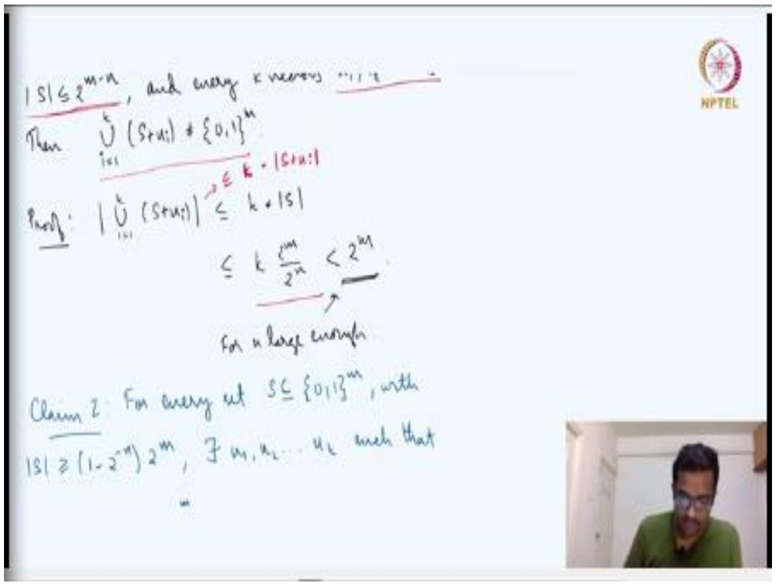
Take any k vectors U 1 to U k, take any k vectors, so S + U 1 S + U 2 all of these contribute to different shifts, all of these are different shifts of S. Now take the union of all these shapes, S + U 1 union S + U 2 etcetera up to S + U k, this will not be the entire universe, this is what it says. This is what I said in pictorially in a notation form, the shifts are not going to cover the universe, something is going to be missed.

And this is going to be proved by merely a counting argument. So, in the right hand side the size is 2 power m 0 1 whole power n and the left hand side it is a union. So, as we know the union of kth sets the maximum size is obtained when the union is all disjoint. There could be intersections between them but then the size of the union only comes down, the size is maximized when the sets are disjoint, so let us take the maximum possible.

So, what is the size of the left hand side? The maximum possible is k times the size of U + i, so k times the, let me just write one more line here, this is at most k times size of S + U i. But then S + U i is just S shifted by something, so size of S + U i is the same as size of S. So, that is why we have k time is size of S. And what is the size of S? We know the size of S is 2 power m divided by 2 power n at most and we know that k is such that n k greater than m.

So, k is such that k = m by n + 1, so that is not going to be enough to make up this. So, k times 2 power m divided by 2 power n will be strictly less than 2 power m. Because k is m divided by n and then you have divided by 2 power n, so that is not going to be enough.

**(Refer Slide Time: 24:48)**



$|S| \le 2^{m-n}$, and every k vectors [illegible]

Then $\bigcup_{i=1}^{k} (S+u_i) \neq \{0,1\}^m$

Proof: $\left|\bigcup_{i=1}^{k} (S+u_i)\right| \overset{\le k \cdot |S+u_i|}{\le} k \cdot |S|$

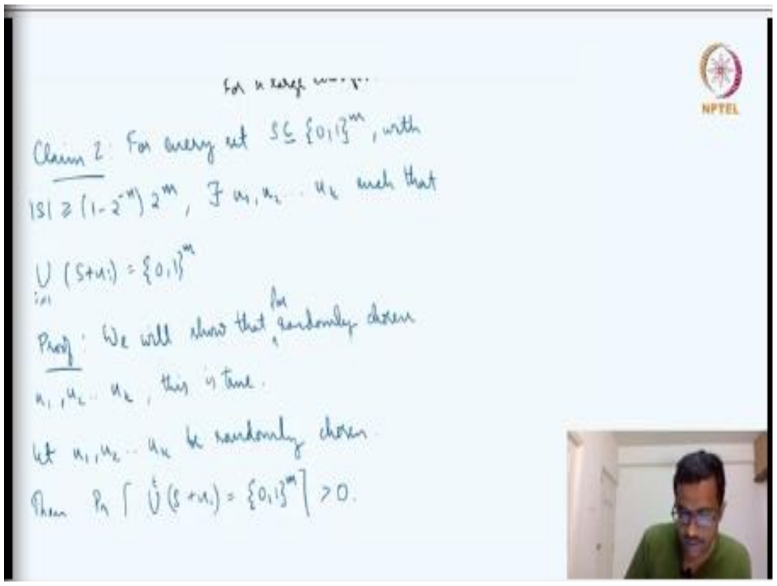
$\le k \frac{2^m}{2^n} < 2^m$

For n large enough

Claim 2: For every set $S \subseteq \{0,1\}^m$, with $|S| \ge (1-2^{-n}) 2^m$, $\exists\, u_1, u_2 \ldots u_k$ such that

So, when n is large enough just by counting argument we show that by shifting S by S into k copies, it is not going to cover the entire universe. Because the size is even in the disjoint case it is not going to be able to cover, that is the first point. The second claim is to show that when S x is large meaning S x is of the size 1 - 1 divided by 2 power n multiplied by 2 power m, there is a way to shift S x k times and cover the entire universe, that is claim 2.

**(Refer Slide Time: 25:38)**



For n large enough.

Claim 2: For every set $S \subseteq \{0,1\}^m$, with $|S| \geq (1-2^{-n})2^m$, $\exists u_1, u_2 \ldots u_k$ such that

$\bigcup_{i=1} (S+u_i) = \{0,1\}^m$

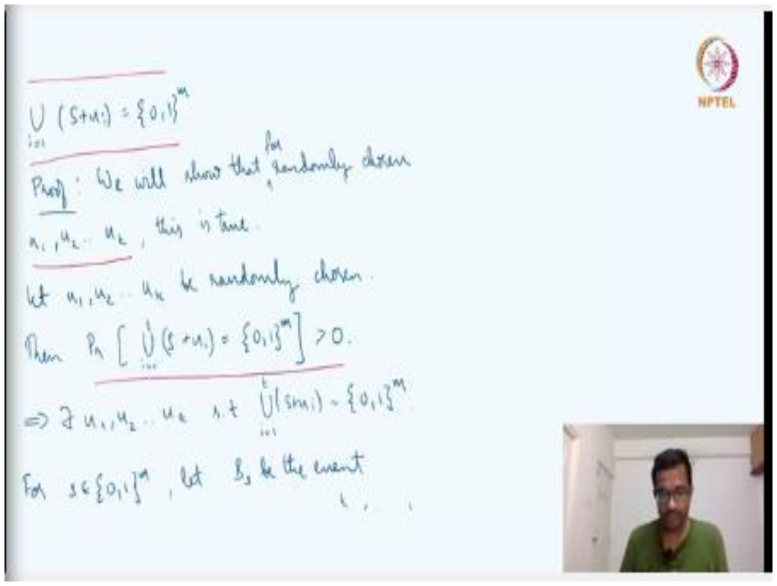
Proof: We will show that for randomly chosen $u_1, u_2 \ldots u_k$, this is true.

Let $u_1, u_2 \ldots u_k$ be randomly chosen.

Then $\Pr[\bigcup^{k} (S+u_i) = \{0,1\}^m] > 0$.

For every set S, again this is just any set S with this size requirement 1 - 1 by 2 power n multiplied by 2 power m there are k U i's or k shifts that will lead to the entire universe being covered with these k shifts.

**(Refer Slide Time: 26:04)**



$\bigcup_{i=1} (S+u_i) = \{0,1\}^m$

Proof: We will show that for randomly chosen $u_1, u_2 \ldots u_k$, this is true.

Let $u_1, u_2 \ldots u_k$ be randomly chosen.

Then $\Pr[\bigcup_{i=1}^{k} (S+u_i) = \{0,1\}^m] > 0$.

$\Rightarrow \exists u_1, u_2 \ldots u_k$ s.t $\bigcup_{i=1}^{k} (S+u_i) = \{0,1\}^m$

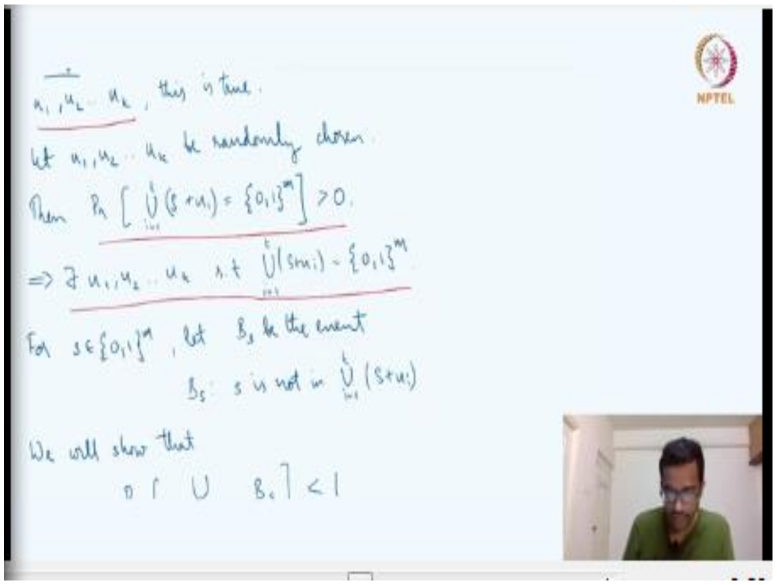
For $s \in \{0,1\}^m$, let $B_s$ be the event

And what we will do in order to show this? is to show that if you choose U i is randomly this will come true. We will show that it is possible even for randomly chosen U i's. So, this is a technique called probabilistic technique? Suppose, so in simple terms if you roll a die the probability of U getting a 1 is 1 over 6. So, the probability of getting a 1 is non zero which means there is some outcome that leads to U getting a 1.

Whereas the probability of U getting a 10 in a single die is 0, that is because there is no possibility. So, what we will show here is for randomly chosen U 1 to U k the probability of this happening is greater than 0. So, if the probability of this happening is greater than 0 that means there is some set of possibilities that lead to it getting a non zero probability, which means there exists some U 1 to U k for which this shifts cover the entire universe.

And which that is enough for us, we do not need to know what they are; all we need to know is the existence of this U 1 to U k. So, what we will show is if U 1 to U k is randomly chosen the probability of shifts covering this universe is strictly greater than 0, this is what we will show.

**(Refer Slide Time: 27:38)**



$\underline{u_1, u_2 \ldots u_k}$, this is true.

Let $u_1, u_2 \ldots u_k$ be randomly chosen.

Then $\underline{\Pr\left[\bigcup_{i=1}^{k}(S + u_i) = \{0,1\}^m\right] > 0}$.

$\Rightarrow \underline{\exists\, u_1, u_2 \ldots u_k \text{ s.t. } \bigcup_{i=1}^{k}(S + u_i) = \{0,1\}^m}$

For $s \in \{0,1\}^m$, let $B_s$ be the event

$B_s$: $s$ is not in $\bigcup_{i=1}^{k}(S + u_i)$

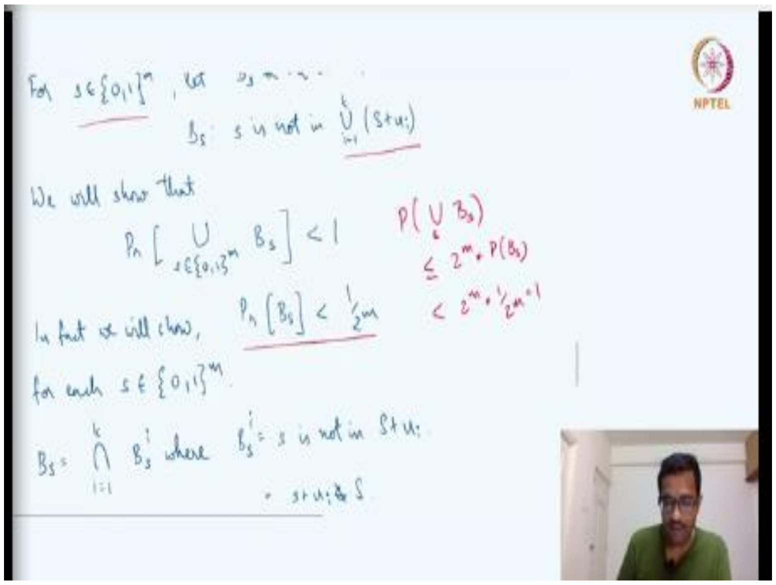
We will show that

$\Pr\left[\bigcup B_s\right] < 1$

For U 1 to U k randomly chosen the probability of the shifts covering U 1 to U k is strictly greater than 0, some non zero quantity. Because if it has to be even let us say 0.01, that means there is some minute set of possibility some U 1 U k that occur with this probability that is giving us this non zero value. So, that is all we need, we need existence of sum U 1 to U k. So, arguing that something happens because the probability of that happening is greater than 0 is called the probabilistic method.

Again this is a very powerful technique that is often used in many proofs in randomized algorithms and also combinatorics and this is our first instance of using it in this course. So, the probability that this is non zero or strictly greater than 0 implies the existence of U 1 to U k which give us the shifts that cover the entire universe. So, provided S is big enough we know the existence of this U 1 to U k.

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



For $s \in \{0,1\}^m$, let [illegible]

$B_s$: s is not in $\bigcup_{i=1}^{k} (S+u_i)$

We will show that

$$P_r\left[\bigcup_{s \in \{0,1\}^m} B_s\right] < 1$$

$$P\left(\bigcup_{s} B_s\right) \leq 2^m \cdot P(B_s) < 2^m \cdot \frac{1}{2^{m+1}}$$

In fact we will show, $P_r[B_s] < \frac{1}{2^m}$

for each $s \in \{0,1\}^m$.

$B_s = \bigcap_{i=1}^{k} B_s^i$ where $B_s^i = s$ is not in $S + u_i$

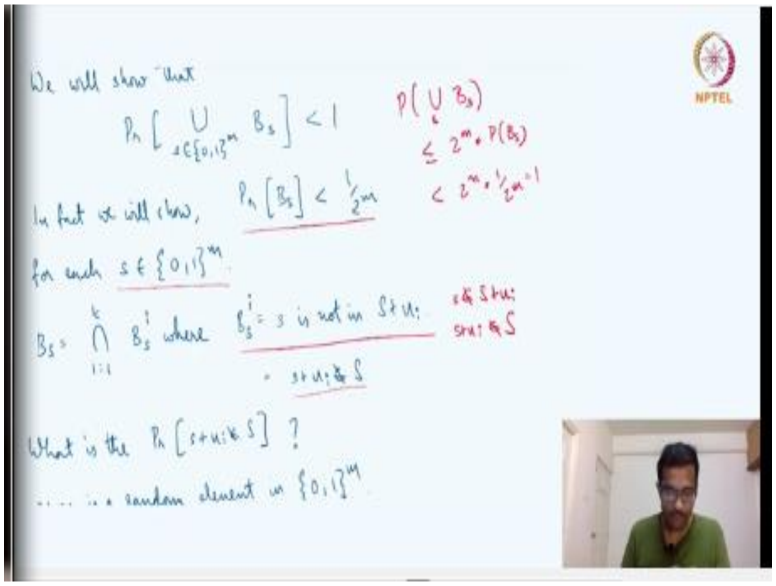
$\equiv s + u_i \notin S$

So, now let us compute the probability. So, the rest of the proof is just going to be estimating this probability and show that this works. So, consider a string s in an m bit long string and let B s be the event that s is not covered by this union of these shifts, B s is the event that s is not covered by the union of these shifts and so we need to show that the union covers everything with probability greater than 0. Instead we will show the negation the complement even has strictly less than one probability. What is the complement event?

The complement event is there is some s that is not covered. So, the union of all any s being not covered, we show that event is happens with probability strictly less than 1. So, in fact again this is also possibly going to be an over estimation. Because it is a union and we will just show that even this, so here again we will use the sum of the probabilities. So, in fact what we will show is that the probability of B s itself, that means probability of s being not covered itself is strictly less than 1 by 2 power n.

So, the probability of their existing or the probability of their existing any s that is not covered is going to be strictly less than 1. So, the property of their x is 1 s is the probability of the union and the probability of the union is at most m times, so probability of the union of all s B s is at most 2 power m times the probability of any B s. And this we know is strictly less than 1 by 2 power m, so it is 2 power m multiplied by 1 by 2 power m which is equal to 1, so it is strictly less than 1. So, now all that we need to show is that for a fixed s the probability that s is not going to be in this shifted union is at most 1 by 2 power m.

**(Refer Slide Time: 31:41)**



We will show that

$$\Pr\left[\bigcup_{s \in \{0,1\}^m} B_s\right] < 1$$

$P(\bigcup_s B_s)$

$\leq 2^m \cdot P(B_s)$

$< 2^m \cdot \frac{1}{2^m} = 1$

In fact we will show, $\underline{\Pr[B_s] < \frac{1}{2^m}}$

for each $\underline{s \in \{0,1\}^m}$.

$B_s = \bigcap_{i=1}^{k} B_s^i$ where $\underline{B_s^i = s \text{ is not in } S + u_i}$

$= \underline{s + u_i \notin S}$

$s \notin S + u_i$

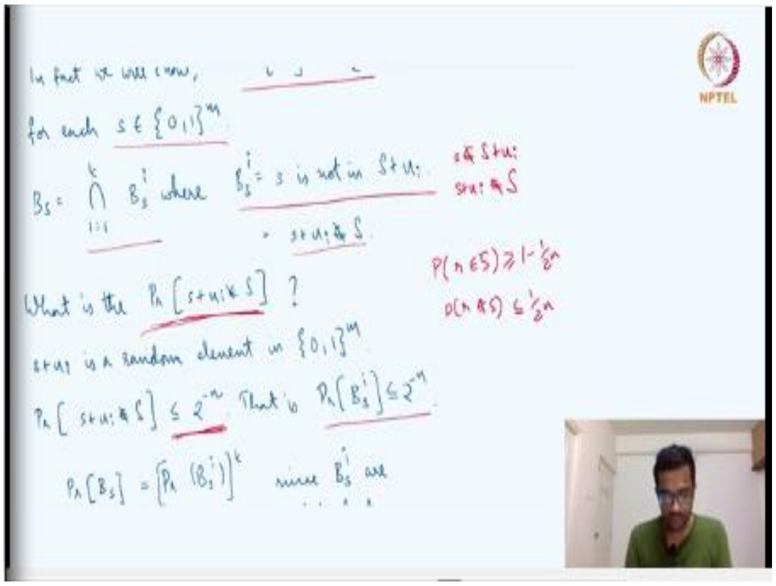
$s + u_i \notin S$

What is the $\Pr[s + u_i \notin S]$ ?

. . . . . . a random element in $\{0,1\}^m$

And this will be shown this be probability of B s is strictly less than 1 by 2 power m will be shown for any m, so s is fixed. So, now we will further try to understand B s, B s is the probability that s is not covered. So, let B s superscript i be the event that s is not in the ith shift small s is not in S + u i. So, s is not in S + u i that is the same as saying s + u i is not in S. Because it is bitwise x r so if you add u i to the s then that is the same as or if you add u i on both sides of the first line you get that s + u i should not be in S. So, B s superscript i is the event that s + u i is not in S which also means that s is not covered in the i th shift. So, what is B s?

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



In fact we will show,

for each $s \in \{0,1\}^m$

$B_s = \bigcap_{i=1}^{k} B_s^i$ where $B_s^i$ = s is not in $S + u_i$ $\quad s \notin S + u_i$ $\quad s + u_i \notin S$

$= s + u_i \notin S$.

$P(n \in S) \geq 1 - \frac{1}{2^n}$

$P(n \notin S) \leq \frac{1}{2^n}$

What is the $\Pr[s + u_i \notin S]$ ?

$s + u_i$ is a random element in $\{0,1\}^m$.

$\Pr[s + u_i \notin S] \leq 2^{-n}$ That is $\Pr[B_s^i] \leq 2^{-n}$.

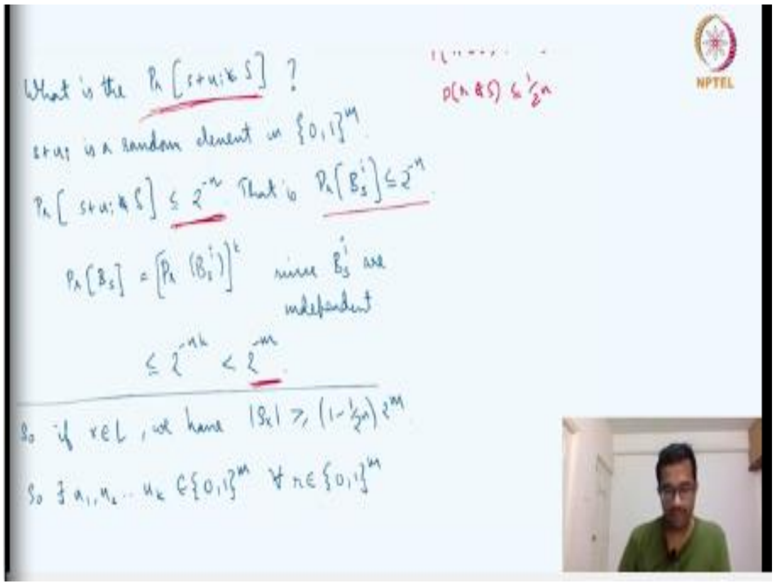
$\Pr[B_s] = [\Pr(B_s^i)]^k$ since $B_s^i$ are

B s is the event that s is not covered by any shift, meaning, it should not be covered by the 1st shift, 2nd shift so on up to kth shifts. So, it is the probability that or it is the event that the S is not covered in any shift, so it is a intersection of all the B s superscript i. So, in fact we will estimate B s superscript i the probability of B s superscript i, what is the probability of B s group superscript i which is same as asking what is the probability of S + u i not being in S.

So, in fact once you shift, so U i is a randomly chosen string and S + U i is also a random element because U i is random. So, wherever you started from you move somewhere randomly, so now this is also a random place. So, what is the probability that s + U i is not in S? We know that S is of size **1** at least 1 - 2 power - n multiplied by 2 power m which means it the probability of it not being there in S is less than or equal to 2 power - 1 by 2 power n.

So, this is because probability of some random element being in S is at least 1 - 1 by 2 power n. So, probability that some random element is not in S is at most 1 by 2 power n, so that means probability of B s i is at most 1 by 2 power n.

**(Refer Slide Time: 34:55)**


What is the $\Pr[s+u_i \notin S]$ ?

$s+u_i$ is a random element in $\{0,1\}^m$

$\Pr[s+u_i \notin S] \leq 2^{-n}$ That is $\Pr[B_s^i] \leq 2^{-n}$

$\Pr[B_s] = [\Pr(B_s^i)]^k$ since $B_s^i$ are independent

$\leq 2^{-nk} < 2^{-m}$

So if $x \in L$, we have $|S_x| \geq (1-\frac{1}{2^n})2^m$

So $\exists u_1, u_2 \dots u_k \in \{0,1\}^m$ $\forall x \in \{0,1\}^m$

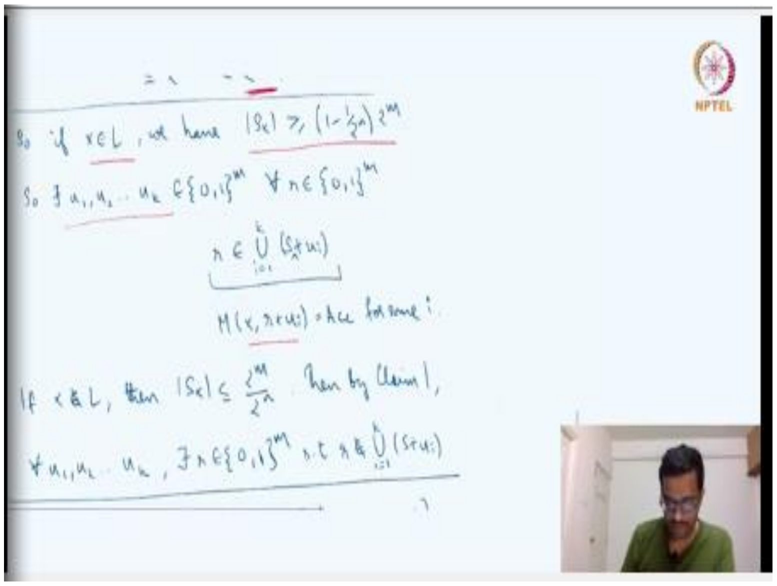
$p(x \notin S) \leq \frac{1}{2^n}$

And this is independent of i because each i is virtually the same because every i, every shift is a randomly chosen U i. So, probability and each of them is independent, so the probability of the intersection the property of B s which is the intersection of all the B s superscript i is nothing but the product because the intersection is over independent probabilities, so it is just the product, so it is 2 power -n whole power k.

And we know that k was chosen to be such that n k is greater than m, so n k greater than m, so -n k is less than -m, strictly less than -m, so we have probability of B s being at strictly less than 2 power –m. And that is all we needed because then the probability of the union is strictly less than 1 which means that there is some shift that covers all the strings S in S and that is what we needed.

So, just to recap the proof we needed to show that there are enough shifts such that all the or we need to show the k is enough such that all the strings are covered. And we show that by showing a random set of k shifts and the probability that the random k shifts do not cover everything is strictly less than 1. And that implies that the probability that the random shifts cover everything is strictly greater than 0. And that is enough to imply the existence of their existence of U 1 to U k which are shifts, such that S + U 1 S + U 2 etcetera S + U k cover the entire universe.

**(Refer Slide Time: 36:53)**



So if $x \in L$, we have $|S_x| \geq (1 - \frac{1}{2^n}) 2^m$

So $\exists u_1, u_2 \ldots u_k \in \{0,1\}^m$ $\forall r \in \{0,1\}^m$

$$\underbrace{r \in \bigcup_{i=1}^{k} (S_x + u_i)}$$

$M(x, r+u_i) = Acc$ for some $i$

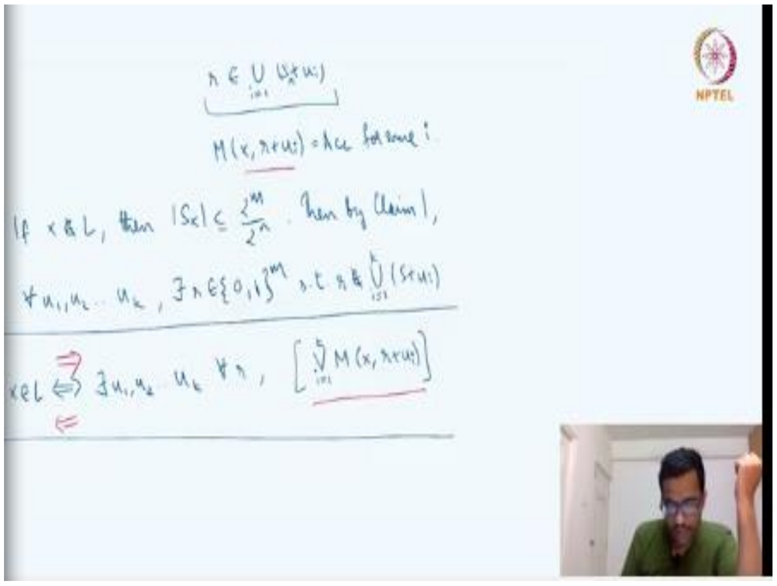
If $x \notin L$, then $|S_x| \leq \frac{2^m}{2^n}$ then by Claim 1,

$\forall u_1, u_2 \ldots u_k$, $\exists r \in \{0,1\}^m$ s.t. $r \notin \bigcup_{i=1}^{k} (S_x + u_i)$

So, what we have is that claim 1 and 2, so first let us see what claim 2 says. If x is in the language S x is big enough to satisfy the requirement of claim 2. And that implies the existence of U 1 to U k, such that for all the strings S is covered in the shifts. So, instead of S let me use the letter r, it is just a letter change. So, for all the r's, r is covered in some shift, the union covers everything. So, any string r any m length string r is in the union, what does this mean?

This means that S x is a set of strings that where x is accepted, remember how S x was defined initially. This is a set of random strings where x is accepted. So, if r is in S x + U i, this means r + U i is in S x, meaning the randomized turing machine m accepts x when guided by r + U i, so because r + U i is an S x. So, if you consider r + U 1, r + U 2, r + U 3 etcetera for some i r + U i will lead x to be accepted. If S x is not in L, so maybe I will just explain this in the same statement in the, so now you may already be seeing how we are coming to the sigma 2 representation.

**(Refer Slide Time: 39:05)**



$\underbrace{r \in \bigcup_{i=1} (S_x + u_i)}$

$M(x, r+u_i) = \text{Acc}$ for some $i$

If $x \notin L$, then $|S_x| \leq \frac{2^m}{2^n}$. Then by Claim 1,

$\forall u_1, u_2 \ldots u_k$, $\exists r \in \{0,1\}^m$ s.t. $r \notin \bigcup_{i=1}^{k} (S_x + u_i)$

$x \in L \overset{\Rightarrow}{\underset{\Leftarrow}{\Leftrightarrow}} \exists u_1, u_2 \ldots u_k \; \forall r, \; \left[ \bigvee_{i=1}^{k} M(x, r+u_i) \right]$

So, same thing let me try to represent x is in L, then there exist U 1, U 2 up to U k which are the shifts, such that for all r in 0, 1 power m either m accepts x and r + U 1 or m accepts x and r + U 2 or m accepts + x and r + U 3 and so on. One of these k things lead to acceptance, that the same thing is saying the r of these k events is true, the r of these k events. So, now you can see why it is a sigma 2 representation?

Because m is a deterministic verifier, deterministic machine, once r is fixed m just performs in a deterministic manner. And the union of k such m's, you can run k m's one by one serially or parallely it is still polynomial time. So, the r operation over k m's is also polynomial time. And you have outside you have their exist followed by for all, so existential and universal quantifiers as sigma 2 would insist, so that is one direction.

Now let us also convince horses of the other direction, just to show, so if and only if now we have shown this direction, when x is not in L also we need to show which is the other direction. Suppose x is not in L then S x is of size 2 power m divided by 2 power n or at most this. By claim 1 whatever U 1, U 2, U k you try, this is what claim 1 said. Any set of k vectors, the shift is not going to cover everything.

So, whatever U 1, U 2, U k you try there will, so it is not going to cover everything means whatever U 1, U 2, U k you try there is going to be some r that is not covered. So, there is for all

U 1 to U k there is an r such that r is not in the shifts. So, which is what we wanted to show, which is the negation of this. So, now with both of this claim 1 and claim 2 together we have shown that x is in L if and only if there x is U 1 to U k, such that for all r this has to be true. The r of m x + r x, r + U i has to be true.

And this is a sigma 2 machine, so we have shown that L has a sigma 2 characterization which is what we set out to do. Just to summarize, so sigma 2 has this characterization. Just to summarize we want to show BPP is in sigma 2 intersection pi 2 but then it we first showed that it is enough to show that BPP is in sigma 2 because BPP is closed under complement. So, we took an arbitrary language in BPP and then we boosted it to such an extent that to set of random strings that lead x to an acceptance covers almost the entire universe 1 - 1 divided by 2 power n fraction.

And if x is not in L the set of strings that lead x to an acceptance is a very, very small portion of the universe. And now the idea is that if x is in the language there are k shifts that you can do that covers entire universe if x is in the language. But if x is not in the language no matter what k shifts you do, it is not going to cover everything. And claim 1 used a bit of probability, again we can go through the details claim 2 used a bit of probability, you can go through the details.

Claim 1 was merely by counting the size of S x and no matter how you place the S x, it is not going to be enough because the size itself is not going to be enough. And with that we get the sigma 2 characterization for L which is what we wanted to show. So, now we have shown that BPP is in sigma 2 and pi 2. And with that we have seen a bunch of randomized algorithms starting from RP, co-RP, ZPP, BPP and so on.

So, now after this we will move to another topic in the next lecture but we will keep coming back to randomized classes. Because we will keep seeing, so just like we saw how BPP relates to sigma 2, we will keep seeing how randomized classes relate to other complexity classes which is the whole point of this course to see how complexity classes relate with each other.