Theory of Computation
Professor Somenath Biswas
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Lecture 26
Pumping Lemma for CFLS
Adversarial Paradigm

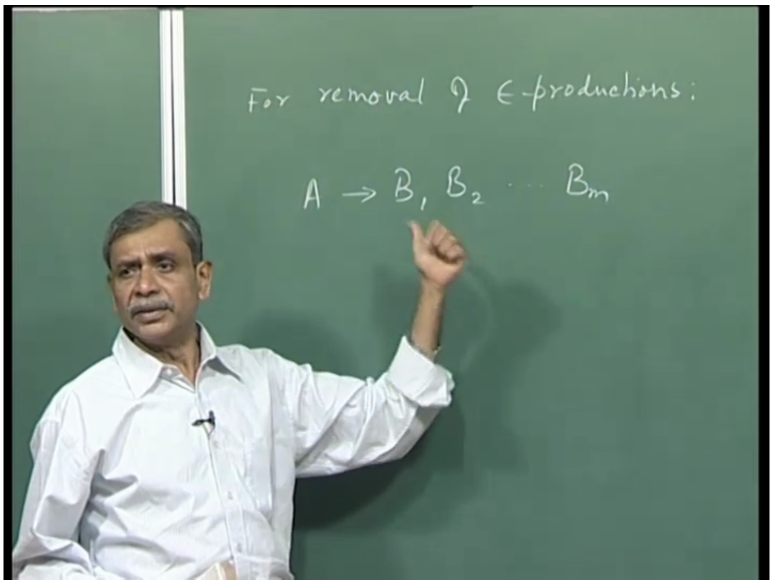
Today we will look at an important result which is the pumping lemma for context free languages using which we will be able to show that certain languages are not context free. Before we state the lemma and prove the lemma and of course make use of the lemma, I would like to make one point which is in connection with what we did last time about getting or transforming a grammar into a Chomsky normal form grammar.

There we had seen how to take a production whose right hand side has you know a long string and then remove such a production and instead in its place we could add productions which had only right hand sides of length two, right?

You recall that was a very simpleidea for getting rid of productions with long, in fact any production whose right hand side had more than two symbols, we could equivalent get productions which will have right hand side only consisting of two symbols and getting rid of such productions with three or more symbols on the right hand side. Now this has one more use and that is if you recall when we considered that our procedure for the removal of epsilon productions, what is the time complexity of this epsilon production removal?

Now unfortunately the way we discuss this, this process this procedure this algorithm can be exponential in the length of the grammar. So very quickly see recall that suppose I had something like B 1, B 2 these are all nonterminals up to B m and it could be that each B i,each of these will be supposenullable.

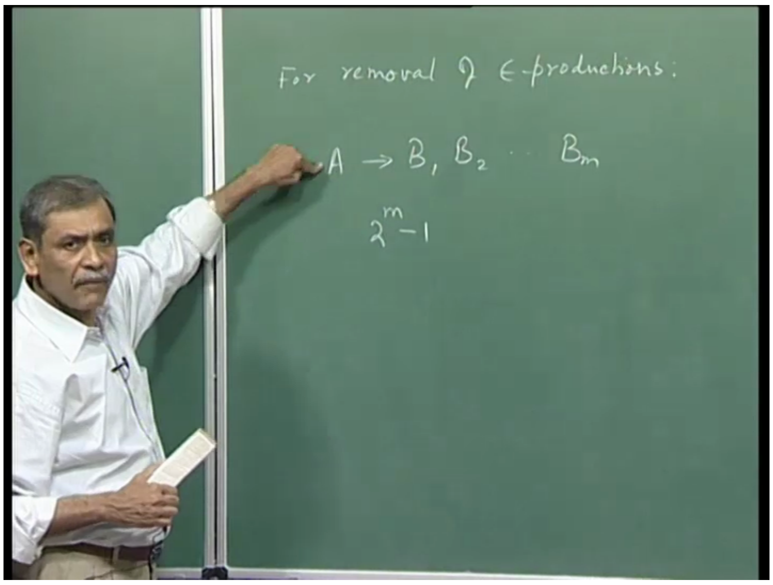
(Refer Slide Time: 03:05)


For removal of ε-productions:
$A \rightarrow B_1 B_2 \cdots B_m$

So in that case how many productions will I get after removing this? I will get,in fact precisely 2 to the power m minus 1 productions we must add in its place, right? Now as you can seeadding so many productions we are adding exponentially many productions in m here which can be of the order of the size of the grammar thatgiven as input to you and therefore just getting rid of such one production and in its place put the requisite number of productions, that itself would be exponential.

So the idea is the way we can makethis algorithm for epsilon productions polynomial time is to first in its place, in the place of this production we should have add first productionswhose right hand side is of length two.

(Refer Slide Time: 04:16)

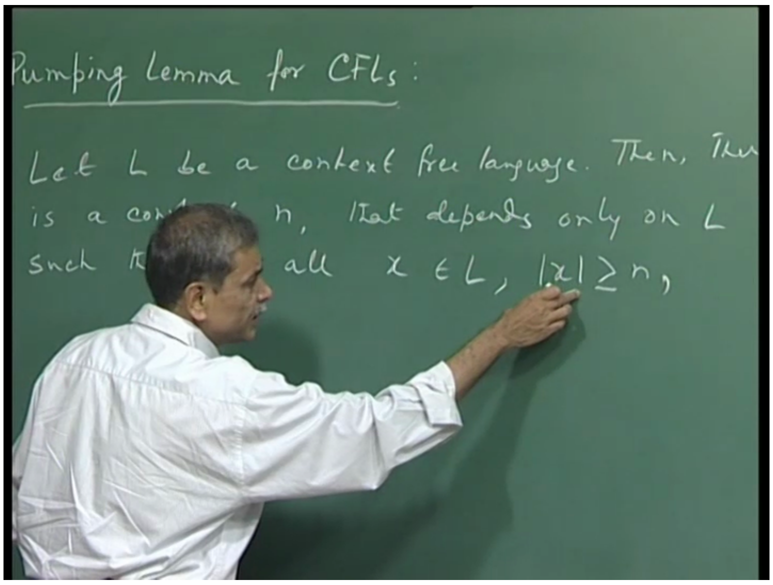

For removal of ε-productions:
$A \rightarrow B_1 B_2 \cdots B_m$
$2^m - 1$

We know how to do it as we discussed Chomsky normal form grammar. And then we apply this process of you know removal of one by one the nullable symbols to get new productions. So in other words from such a production if I can always get equivalently productions whose right hand side has only exactly two symbols. And how many such productions I will need to add? Only linear remaining, right? And then we do what we did for removal of epsilon productions, right?

So in that case you should realize, it is not difficult to see, then our procedure is going to be polynomial time, right? Now let us get back to pumping lemma for context free languages. Now you will remember corresponding pumping lemma for regular languages and the format of the lemma is very similar, right? Sothe lemma goes like this that let L be a context free language.

Then there is a constant n that depends only on L such that for all x in L with length of x greater than equal to n,that is important and the other thing that we are saying isthese strings have large enough length.

(Refer Slide Time: 07:12)

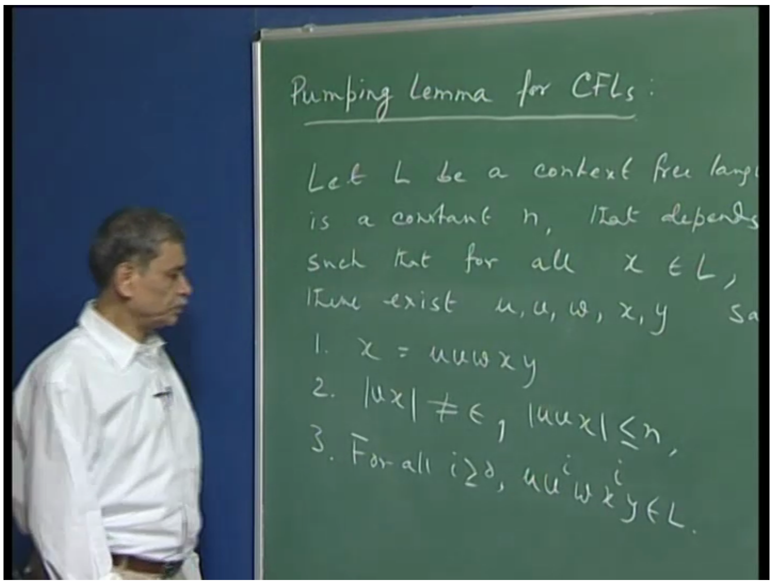


Pumping Lemma for CFLs:

Let L be a context free language. Then, there is a con n, that depends only on L such th all $x \in L$, $|x| \geq n$,

There exists u v wx y satisfying, now we give a number of conditions. First condition is that the string x is the concatenation of these five strings. Second condition is this string v x is not empty. This is another way of saying at least one of the two strings v and x must be non-empty. And the third, for all i greater than equal to 0, u v i w x i y is also in the language. Andwe will qualify one more thing thatnot only v x is non-empty, we will also say aboundon this v x which will say that the length of u v x is bounded by n, okay.

(Refer Slide Time: 08:58)



Pumping lemma for CFLs:

Let L be a context free lang

is a constant n, that depends

such that for all $x \in L$,

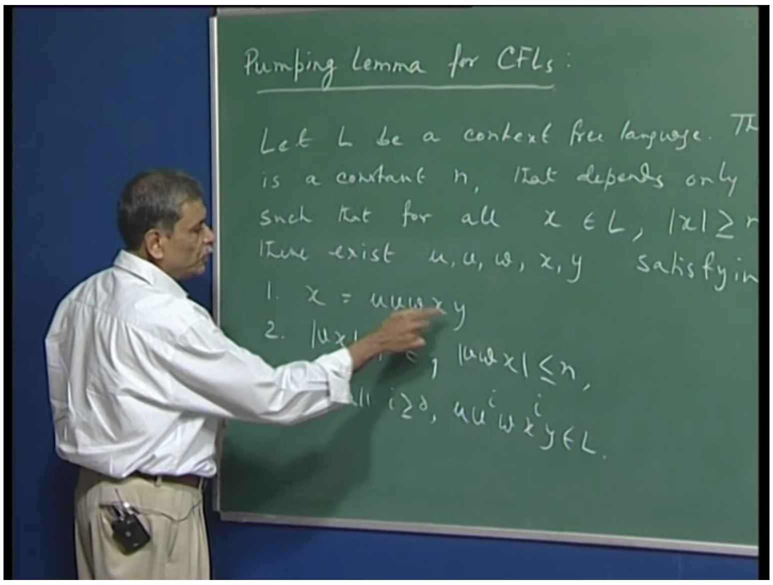
there exist u, v, w, x, y sa

1. $x = uvwxy$
2. $|vx| \neq \epsilon$, $|vwx| \leq n$,
3. For all $i \geq 0$, $uv^iwx^iy \in L$.

So first of all just let us read the lemma, what is it saying? Before we make use of lemma or we prove the lemma, the conditions what it is saying. So it is saying that let L be any context free language then you can find a constant n which will depend only on L. If you recall that is exactly how we started the pumping lemma for regular languages also.

Now it says that for all x in the language which are large enough that means whose length is greater than this n this constant, you will be able to find or there you can find or one can find u v w x y and that will satisfy the following. One is that this string x is the concatenation of these five strings. Then v x is non-empty. That means one of v or one of x is non-empty. Also it says that, I am sorry I should have said v w x, right. This part of the string, the middle three v w x together,their length is bounded by n, right?

(Refer Slide Time: 10:29)

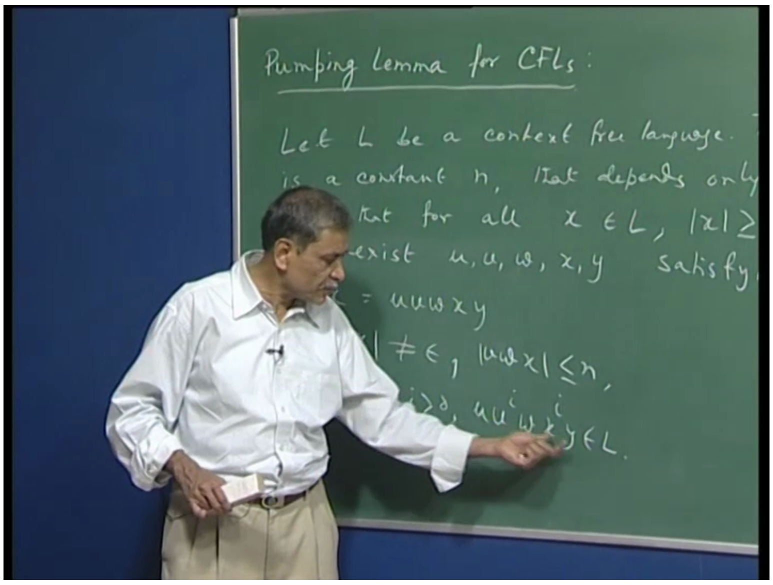


Pumping Lemma for CFLs:

Let L be a context free language. Th
is a constant n, that depends only
such that for all $x \in L$, $|x| \geq n$
there exist u, v, w, x, y satisfyin
1. $x = uvwxy$
2. $|vx|$ ..., $|vwx| \leq n$,
3. ...ll $i \geq 0$, $uv^iwx^iy \in L$.

So actual lengths can be very large. However v w x, that part is bounded by n, its length is bounded by n. And this is where the pumping occurs. You can either pump down by making i equal to 0 or you can pump up by making i greater than equal to 2. What it is saying is that u v to the power i that means i copies of v followed by w, theni copies of x followed by y.

(Refer Slide Time: 11:09)



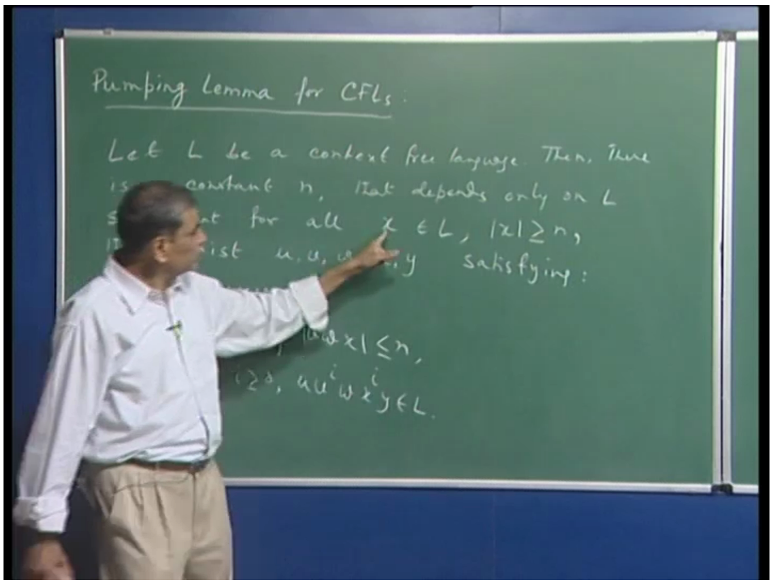
Pumping Lemma for CFLs:

Let L be a context free language.
is a constant n, that depends only
...that for all $x \in L$, $|x| \geq$
...exist u, v, w, x, y satisfy
$= uvwxy$
$| \neq \epsilon$, $|vwx| \leq n$,
$i \geq 0$, $uv^iwx^iy \in L$.

So this is another string and we are claiming that this thing also will be in the language. So the long and short of the whole thing is that if you take alarge enough string in the language then such a string can be pumped. And what you pump is ofcourse non-empty and other thing it is saying that the portion where the pumping will occur, that portion is not too long. Just remember n is a constant for language L.

And now one point often you know strikes peoplewhich really means, see from this x how many new such strings that we are talking of which will be in the language L?

(Refer Slide Time: 12:02)



Pumping Lemma for CFLs:

Let L be a context free language. Then, there is [illegible] constant n, that depends only on L s[illegible]t for all $x \in L$, $|x| \geq n$, [illegible]ist u, v, [illegible] y satisfying:

[illegible] $|vwx| \leq n$,

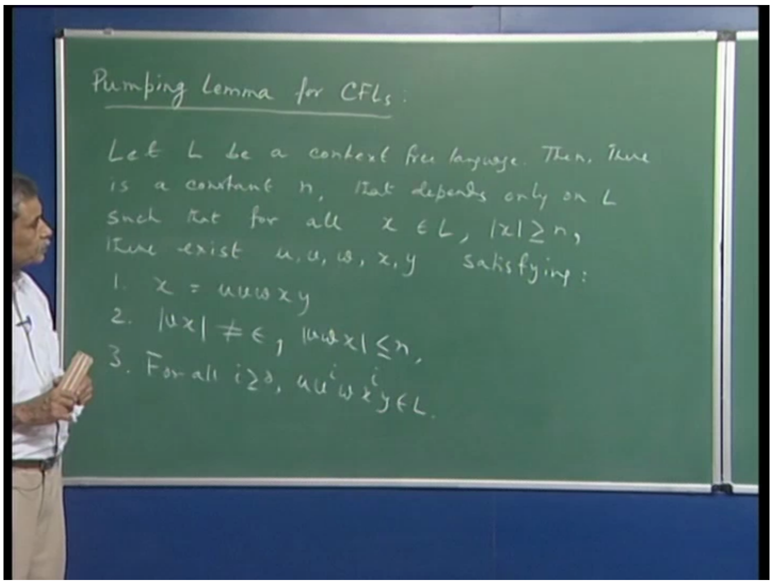
[illegible] $i \geq 0$, $uv^iwx^iy \in L$.

We are saying for all i greater than equal to 0, u v i w x i yall these strings are in the language. How many such strings are there? Clearly infinitely many. So we are asserting that all these infinitely many strings are in the language L. But that does not mean that all context free languages are infinite. So let meput it this way. We know that every finite language is regular. We also know that every regular language is context free. Therefore every finite language is also a context free language.

If you recall long back right when we started the discussion on context free grammars we showed how to take a machine aDFA for a regular language and then from that DFA we can obtain a context free grammar which will generate the same language. Thereby we prove that every regular language is also a context free language. And ofcourse every finite language is regular. So therefore every finite language is context free.

And now it seems we are saying that if L is a context free language and somewhere down below you are saying that all these strings, infinitely many strings are inthat same language, you should be careful to see why it is not a ruling out the possibility of languages which are context free which can be finite. Although we are stating this lemma. I leavethat as an exercise.

(Refer Slide Time: 14:06)



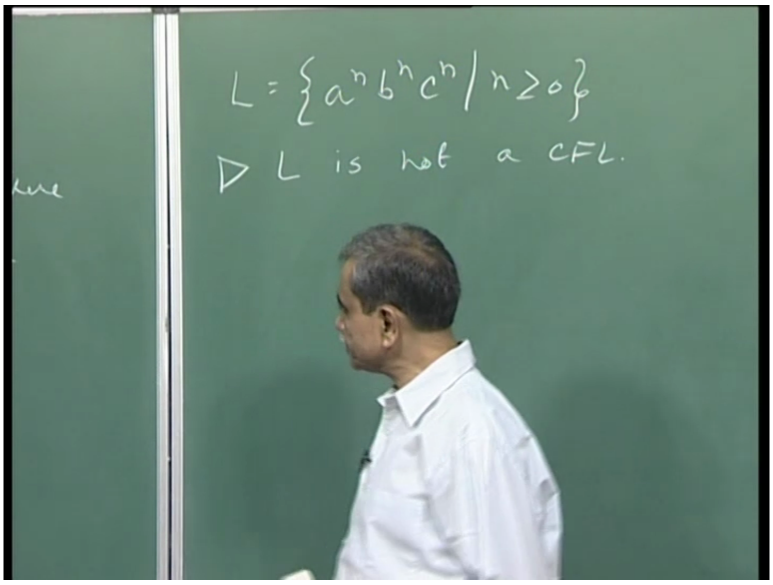
Pumping Lemma for CFLs:

Let L be a context free language. Then, there is a constant n, that depends only on L such that for all $x \in L$, $|x| \geq n$, there exist u, v, w, x, y satisfying:

1. $x = uvwxy$
2. $|vx| \neq \epsilon$, $|vwx| \leq n$.
3. For all $i \geq 0$, $uv^i w x^i y \in L$.

Before we prove the lemma let me quickly give one example of the use of this lemma to show that some language is not context free. In fact the language we will consider is this and let us prove assuming this lemma to be true that L is not a context free language, okay.

(Refer Slide Time: 14:51)



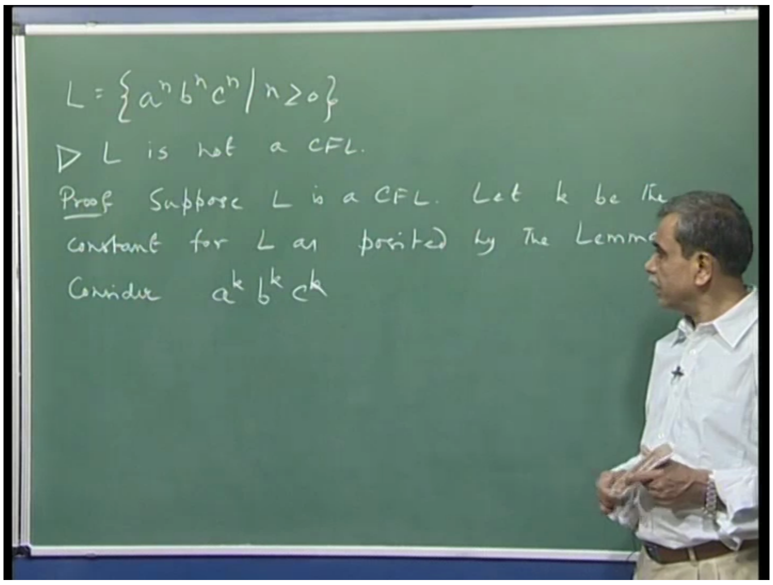
$L = \{a^n b^n c^n / n \geq 0\}$

▷ L is not a CFL.

Now the way this proofs go is by contradiction. So we will say that suppose we are giving the proof. Suppose L is a CFL. Then what isthe lemma saying? That if it is a context free language then there is certain constant n which depends only on the language L,etc. So we will say that let k be the constant for L as posited by the lemma, okay. Now the lemma says that if you take any string whose length is greater than k.

So let me take such a string. Consider the string a k b k c k. Trivially this string is of length three k, I mean therefore the lemmafor all those conclusions will be applicable for this particular string, right?

(Refer Slide Time: 16:41)



$L = \{a^n b^n c^n / n \geq 0\}$

▷ L is not a CFL.

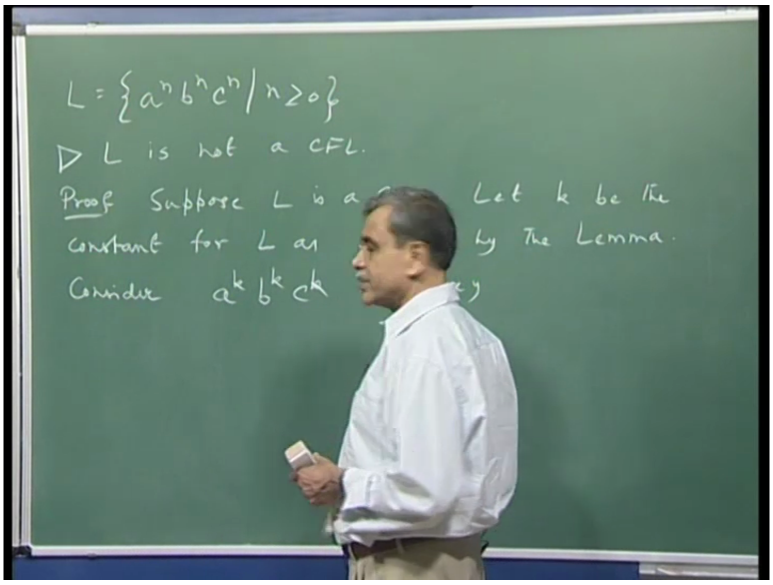
<u>Proof</u> Suppose L is a CFL. Let k be the constant for L as posited by the Lemm

Consider $a^k b^k c^k$

Then what is the lemma stating? Thatthere will exist or we should be able to find out u v w x y in the string such that those things will apply. Now where will these u v w x y occur in the string? Can I assume let u be this, v be this, w be this,etc. I cannot assume that because for the application of the lemma, you see ultimately what I want to do is to prove that L is not a CFLand get some string which is not of this form a n b n c n.

And for thatI should be prepared to handle any break up of this string into u v w x y. We will come to that point a little later. We will elaborate that a little later. But right now you see to apply this lemma I have tosay that I know that there exist u v w x y whose concatenation is this string.

(Refer Slide Time: 18:04)



$L = \{a^n b^n c^n \mid n \geq 0\}$

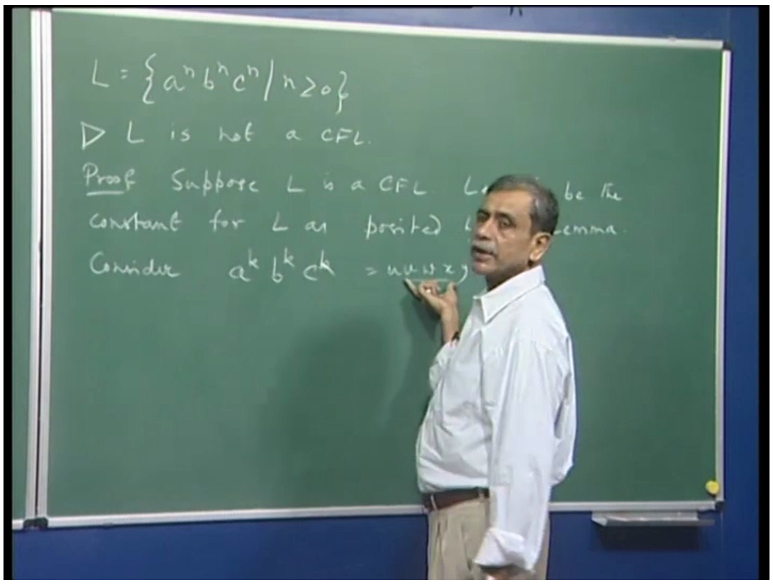
▷ L is not a CFL.

Proof Suppose L is a Let k be the constant for L as by the Lemma.

Consider $a^k b^k c^k$

Further I know two things that v w x, this part, its length is bounded by the pumping lemma constant. So in this case what we can certainly say that this v w x part, its length is bounded by the constant k here which is the pumping lemma constant for this language. So where can this v w x lie?

(Refer Slide Time: 18:32)



$L = \{a^n b^n c^n \mid n \geq 0\}$

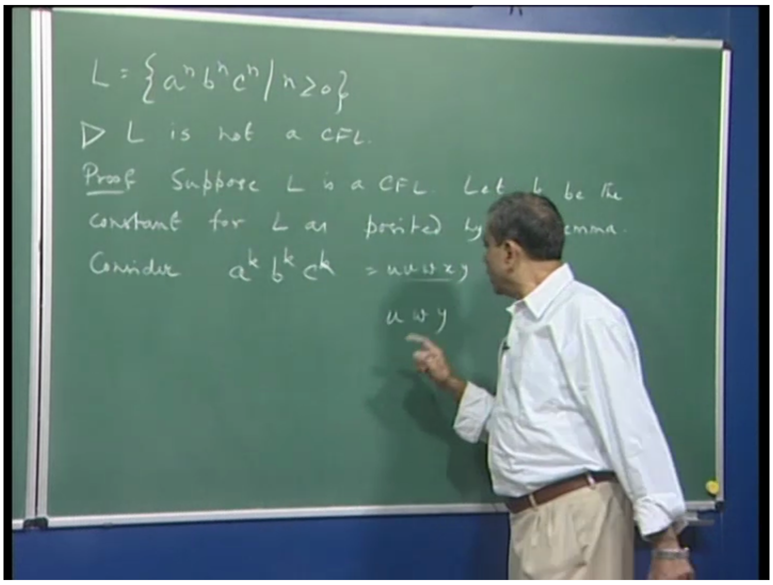
▷ L is not a CFL.

Proof Suppose L is a CFL. L be the constant for L as posited Lemma.

Consider $a^k b^k c^k = uvwxy$

It can of course lie entirely within the a, right? It can straddle some part of a and some part of b. Remember that v w x is a contiguous string. So it is kind of a window of length k or less.That is what we are seeking. So that window can be entirely within a or it can be part of a followed by a part of b and soon. Entirely in b, part of b, part of c, entirely in c. These are the cases, right? Now you see that suppose v w x was entirelywithin a. And now consider i is equal to 0 to get this string u w y.

(Refer Slide Time: 19:29)



$L = \{a^n b^n c^n \mid n \geq 0\}$

▷ L is not a CFL.

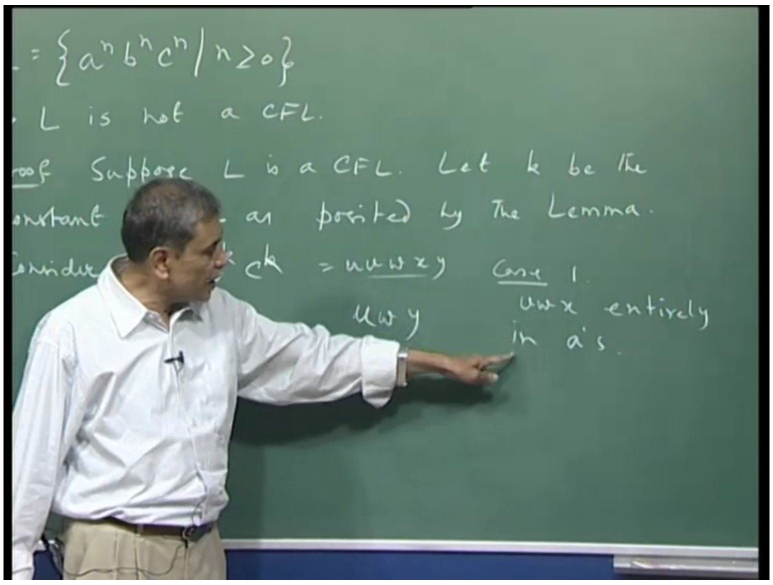
Proof Suppose L is a CFL. Let k be the constant for L as posited by emma.

Consider $a^k b^k c^k = u\underline{vwx}y$

uwy

The lemma states that both v and x cannot be empty. Now we are considering the case 1 that v w x entirely in a's. So after taking out v and x, the string that result is indeed u w y and then this string u w y which I obtained from this string whichwas a to the power k, b to the power k, c to the power k. I have removedv and x which is non-empty and that was entirely in a's.

(Refer Slide Time: 20:15)



$= \{a^n b^n c^n \mid n \geq 0\}$

L is not a CFL.

roof Suppose L is a CFL. Let k be the

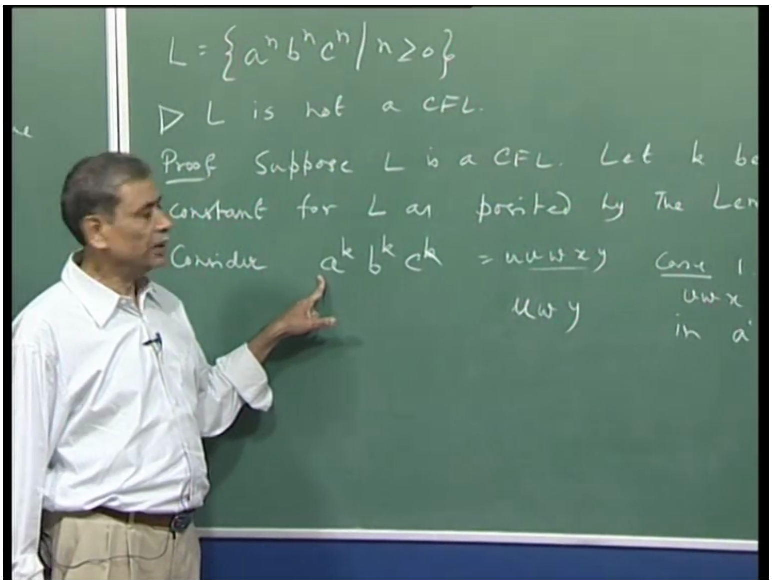
onstant as posited by The Lemma.

onsider $c^k = u\underline{vwx}y$ Case 1.

uwy vwx entirely in a's.

So the long and short of this whole thing is that in this string number of a's is less than in this string clearly.

(Refer Slide Time: 20:25)



$L = \{a^n b^n c^n / n \geq 0\}$

▷ L is not a CFL.

Proof Suppose L is a CFL. Let k b

constant for L as posited by the Len

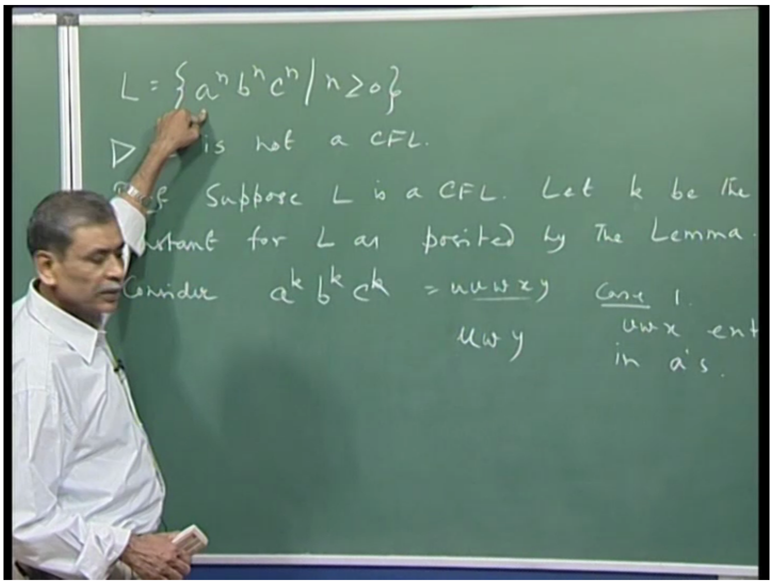
Consider $a^k b^k c^k = u\underline{vwx}y$ Case 1

uwy uwx

in a'

So in that case the number of a’s is less and nothing has changed because so far as b’s and c’s are concerned. Becauseyou have removed v and x but that by assumption will not have either b’s or c’s. So therefore in this string u w y you have less number of a’s than b’s and c’s, okay. So take the next case. So that means what? In this case what we are getting a string which is not of this form and therefore that string is not in the language L.

(Refer Slide Time: 21:03)



$L = \{a^n b^n c^n / n \geq 0\}$

▷ is not a CFL.

f Suppose L is a CFL. Let k be the

stant for L as posited by the Lemma.

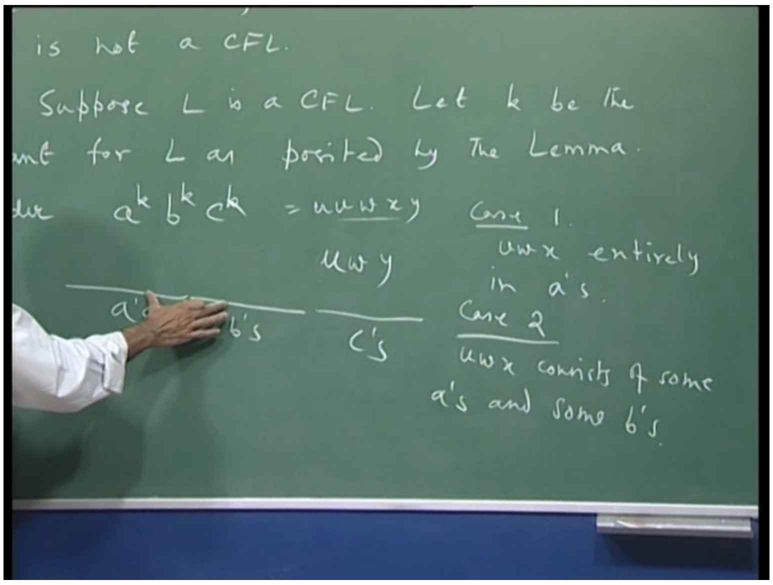
Consider $a^k b^k c^k = u\underline{vwx}y$ Case 1.

uwy uwx ent

in a's.

So what we can say? That clearly this case 1 could not have happened, alright? So okay let us see another case. Remember there are five such cases. Case 2 is that v w x consists ofsome a’s and some b’s, okay. So remember the picture you should keep in mind. So there are some number of a’s followed by some number of b’s followed by some number of c’s. And this number is k and that is the pumping lemma constant and your v w x,this part has to be of length at most k. So either that v w x part is here or here.

(Refer Slide Time: 22:08)



is not a CFL.

Suppose L is a CFL. Let k be the

nt for L as posited by the Lemma.

der $a^k b^k c^k = u\underline{vwx}y$

uwy

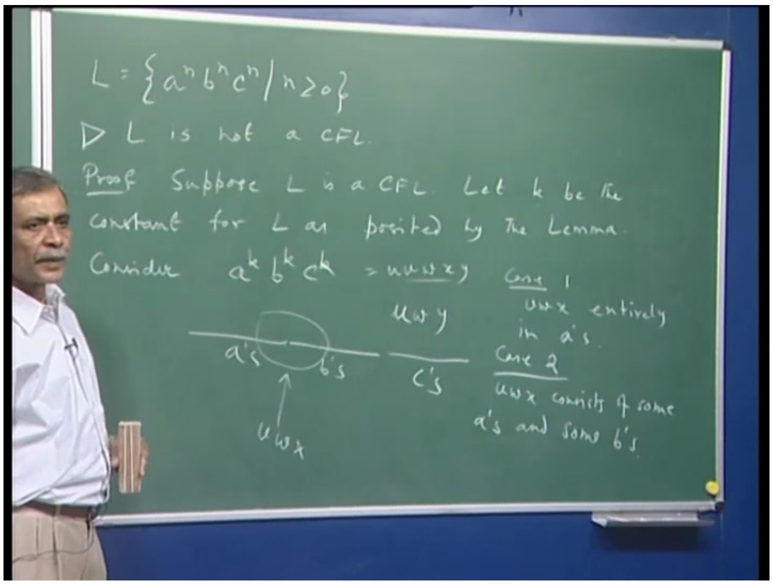
a's b's c's

Case 1. vwx entirely in a's.

Case 2 vwx consists of some a's and some b's.

This is the case now we are considering, right? So let us say somewhere here you had your v w x, okay.

(Refer Slide Time: 22:18)



$L = \{a^n b^n c^n \mid n \geq 0\}$

▷ L is not a CFL.

Proof Suppose L is a CFL. Let k be the constant for L as posited by the Lemma.

Consider $a^k b^k c^k = u\underline{vwx}y$

uwy

a's b's c's

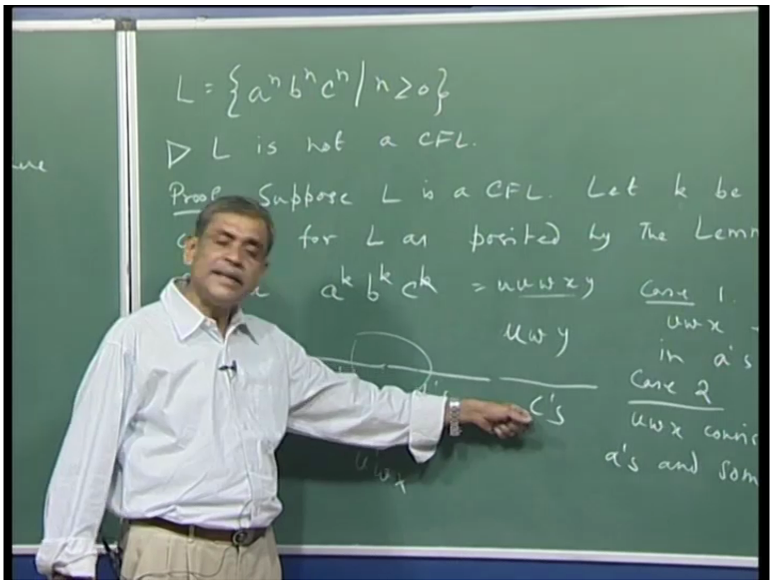
vwx

Case 1 vwx entirely in a's.

Case 2 vwx consists of some a's and some b's.

Now again both v and x cannot be empty. So therefore what? One thing is certain that both v and x could contain likev could contain a’s and x could contain b’s and so on. But surely because v w x part is here, this part does not contain any c’s.

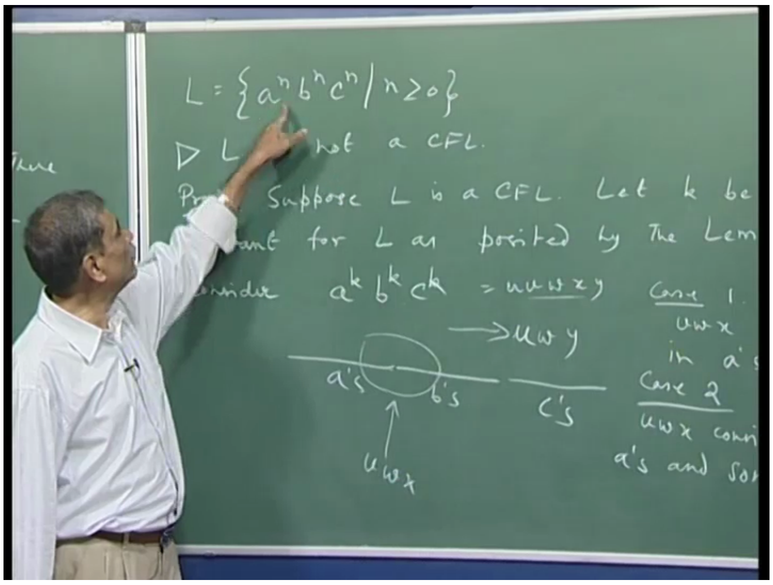
(Refer Slide Time: 22:48)



So now when you remove v and x to get the same kind of string u w y as before, this could have lessened the number of a's and the number of b's or only a's or only b's. But certainly the number of c's in u w y will remain k as in this string. So what has happened that some number of a's possibly and or some number of b's possibly have been removed, right? But the number of c's has not changed.

So the new string that you obtain after pumping down is a string which has more c's than a's and b's, right? So this case 2 also could not have occurred because in that case we would obtain a string which is not of this form and therefore not in the language.

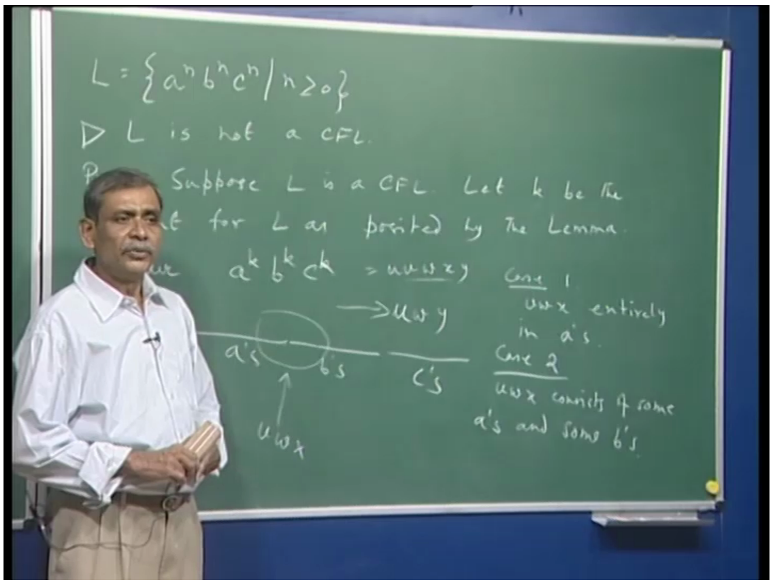
(Refer Slide Time: 23:50)


L = {a^n b^n c^n / n ≥ 0}
▷ L is not a CFL.
Proof Suppose L is a CFL. Let k be
constant for L as posited by the Lem
Consider a^k b^k c^k = uvwxy
Case 1. uwx in a's
→ uwy
a's b's c's
Case 2 uwx conta
a's and so
uwx

See that is out of the question because we have assumed L is in a CFL. So all the strings that I get after doing the pumping up or down should be in the same language L, right? So this way now I consider the other three cases. That is case 3 was you consider the case that v w x part is entirely in b’s, right?

Then fourth case is v w x part straddles both some b’s and some c’s and the last case is the v w x part straddles ordoes not straddle two b’s and c’s. It only consists of c’s. Now you see all these other three cases are also very similar to either case 1 or case 2.

(Refer Slide Time: 24:48)

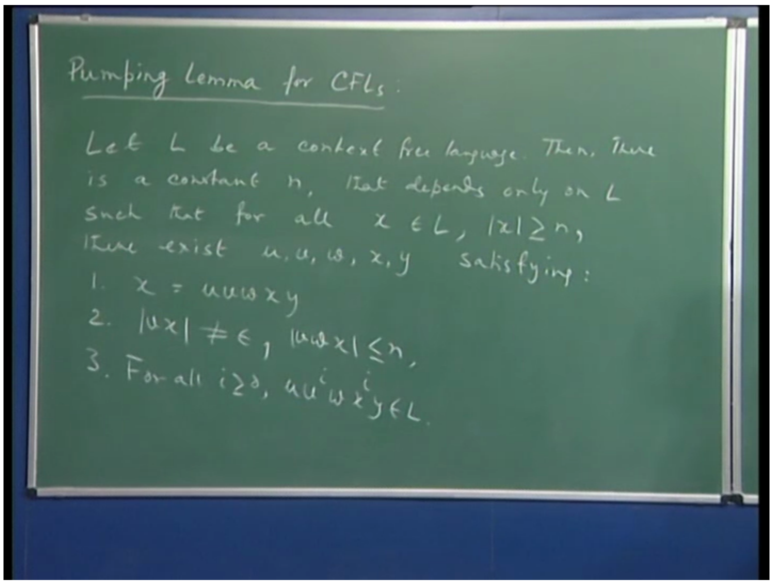


$L = \{a^n b^n c^n / n \geq 0\}$
▷ L is not a CFL.
Suppose L is a CFL. Let k be the
for L as posited by the Lemma.
$a^k b^k c^k = uvwxy$
$\rightarrow uwy$
Case 1
vwx entirely in a's.
Case 2
vwx consists of some a's and some b's
a's
b's
c's
vwx

In that case what is the conclusion? That whatever you do, wherever you consider this break up of u v w x y, in each of these cases what we find is that we are able to obtain a string using the pumping lemma which is not of this form. But that will contradict the lemma and therefore we are coming to a contradiction. So that means what? Our initial assumption must be false or initial assumption was L is a CFL. So we have proved that L is not a CFL because if you assume L is a CFL then you are contradicting the lemma which of course is true.

We saw an application of the lemma through which we proved a certain language is not context free. And all the applications of the lemma will be of this kind. That we will prove that some language to be not context free. We should be quite clear the way this proof went and that we will be,provided we understand the structure of this statement a little more clearly.

(Refer Slide Time: 26:13)



Pumping Lemma for CFLs:

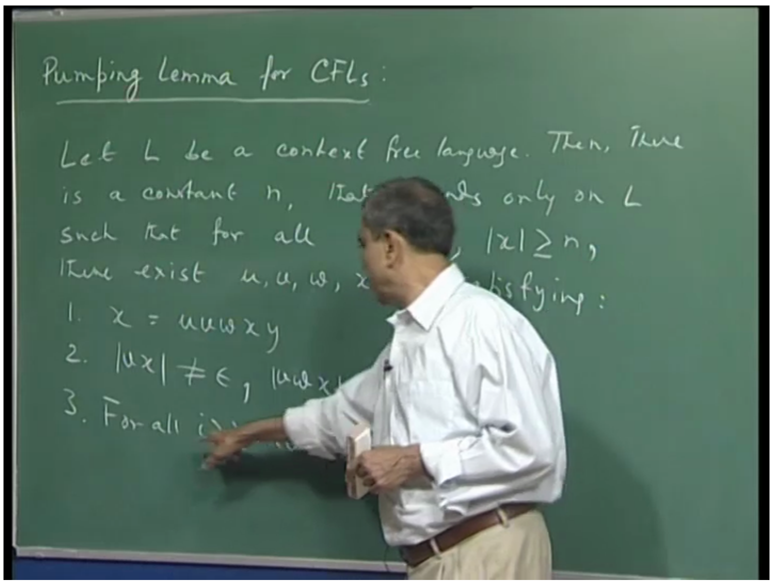
Let L be a context free language. Then, there is a constant n, that depends only on L such that for all $x \in L$, $|x| \geq n$, there exist u, v, w, x, y satisfying:

1. $x = uvwxy$
2. $|vx| \neq \epsilon$, $|vwx| \leq n$,
3. For all $i \geq 0$, $uv^iwx^iy \in L$.

You seethere are several quantifiers, existential as well as universal. Whatit is saying that given that L is acontext free language, first of all it is saying there is an n which we have curly the pumping lemma constant, right? Then it is saying what? That for all x in L, so after this existential quantifier there is a universal quantifier. For all x in L and length of x is greater than n, right?

For all x in L and length of x being greater than n. Then what? Then there is existential quantifier again. So let me write it there exist u v w x y, right? After that what we have? Againyou know a universal quantifier for all i.

(Refer Slide Time: 27:47)



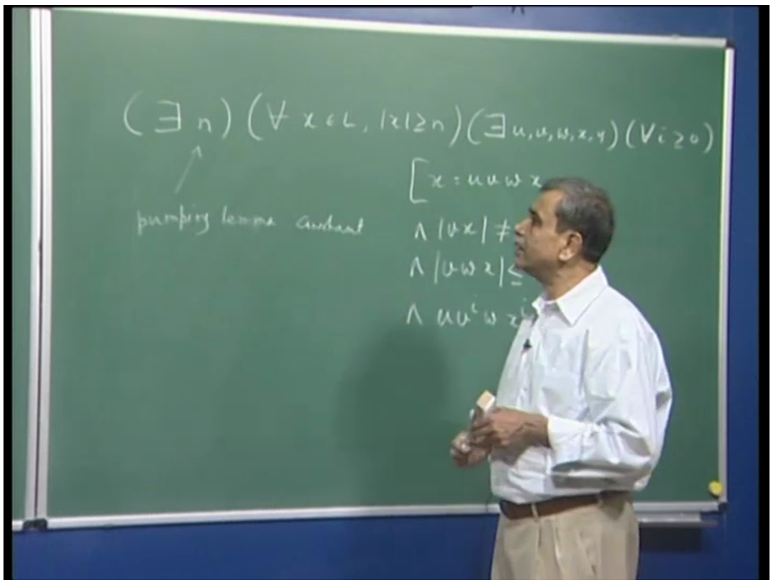
Pumping Lemma for CFLs:

Let L be a context free language. Then, there is a constant n, that ... ds only on L such that for all ... $|x| \geq n$, there exist u, v, w, x ... isfying:

1. $x = uvwxy$
2. $|vx| \neq \epsilon$, $|vwx$ ...
3. For all i ...

Then for all i greater than equal to 0. And this part is the rest of the statement. So basically it is saying that the string x is u v w x y and we are saying that v x is not empty andwe are also saying that v w x, this length is bounded by n, right? And we are saying that u v iw x i y, this is the language, right? This is the logicor you knowif I use the language of first order logic, this is howwe state that lemma.

(Refer Slide Time: 28:56)



$(\exists n)(\forall x \in L, |x| \geq n)(\exists u, v, w, x, y)(\forall i \geq 0)$

$\overset{\uparrow}{\text{pumping lemma constant}}$

$[x = u v w x$

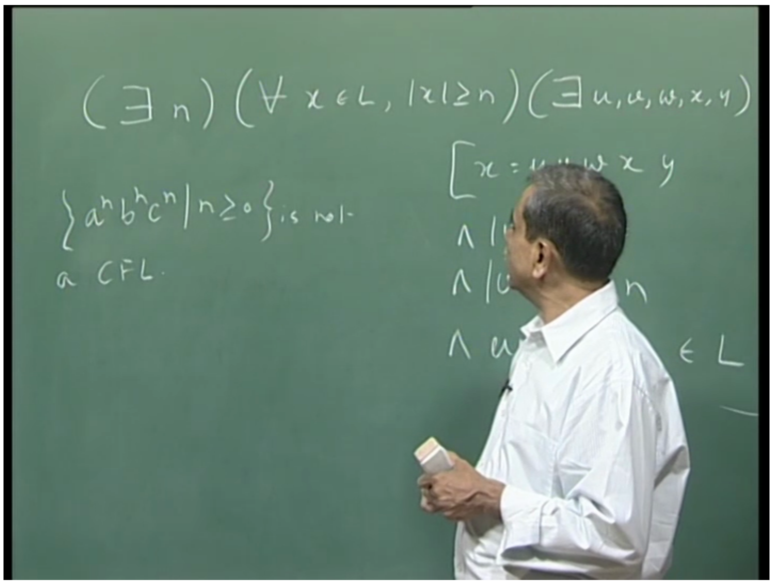
$\wedge\ |v x| \neq$

$\wedge\ |v w x| \leq$

$\wedge\ u v^i w x^i$

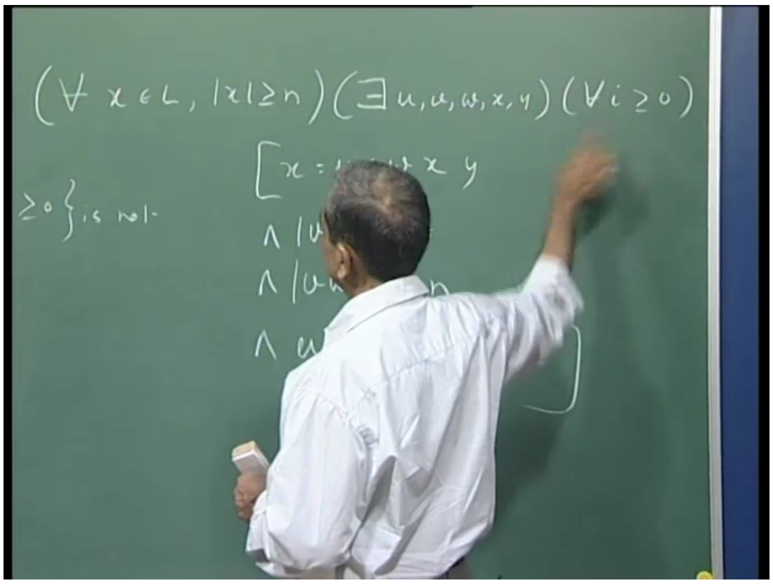
The reason for writing like this is to make ourselves clear that in the application, you see, so just go back to the example that we had given. We wanted to show that the language a n b n c n, this language is not a CFL, okay. Now as we mentioned in the case of pumping lemma for regular languages, you knowit is useful to think of this proof that this language is not context free. Proof of this as a game between you, the prover and an adversary.

(Refer Slide Time: 29:56)


$(\exists n)(\forall x \in L, |x| \geq n)(\exists u,v,w,x,y)$
$\{a^n b^n c^n \mid n \geq 0\}$ is not a CFL.
$[x = uvwxy$
$\wedge$
$\wedge |v$ n
$\wedge u$ $\in L$

Recall in that game aswell as in this game you see we alternate. Adversary does something and then I do something and then finally something is evaluated. Then either I win or the adversary wins. So always remember that prover, the one who wishes to show that some language is not context free, that person I am calling prover, he has complete freedom on the universal quantifiers. Sohis move comes when the universal quantifier is being considered. So like his move is here,his move is here.

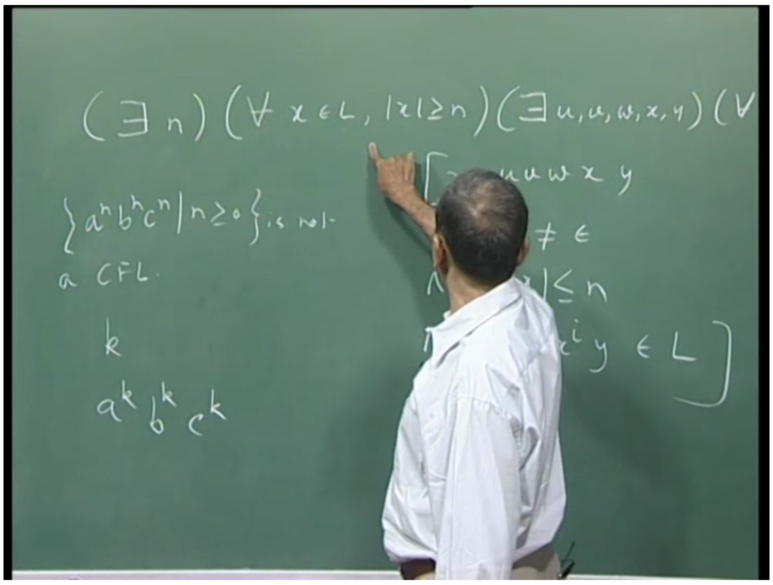
(Refer Slide Time: 30:52)


$(\forall x \in L, |x| \geq n)(\exists u,v,w,x,y)(\forall i \geq 0)$
$\geq 0\}$ is not
$[x = uvwxy$
$\wedge |v$
$\wedge |v$
$\wedge u$

So in other words if you think of what we did, we did not assume what is the pumping lemma constant, is it. So whatever be the pumping lemma constant. That means we are assuming let the adversary give me any constant. Depending he gave the constant k. If you recall we said that k is the pumping lemma constant. Then comes my turn,me in the sense the prover. So now I can choose any x.

So in order to win the game I choose the string x as a k b k c k. And this string has to be in the language andits length should be greater than the pumping lemma constant that is okay. SoI as the prover choseor you know moved legally this part.

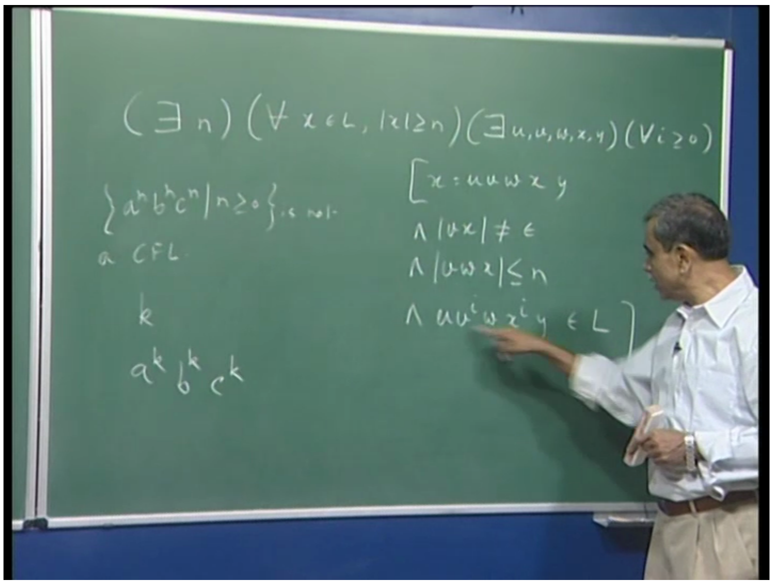
(Refer Slide Time: 31:59)



$(\exists n)(\forall x \in L, |x| \geq n)(\exists u, v, w, x, y)(\forall$
$uvwxy$
$\neq \epsilon$
$| \leq n$
$y \in L]$
$\{a^n b^n c^n \mid n \geq 0\}$ is not a CFL.
k
$a^k b^k c^k$

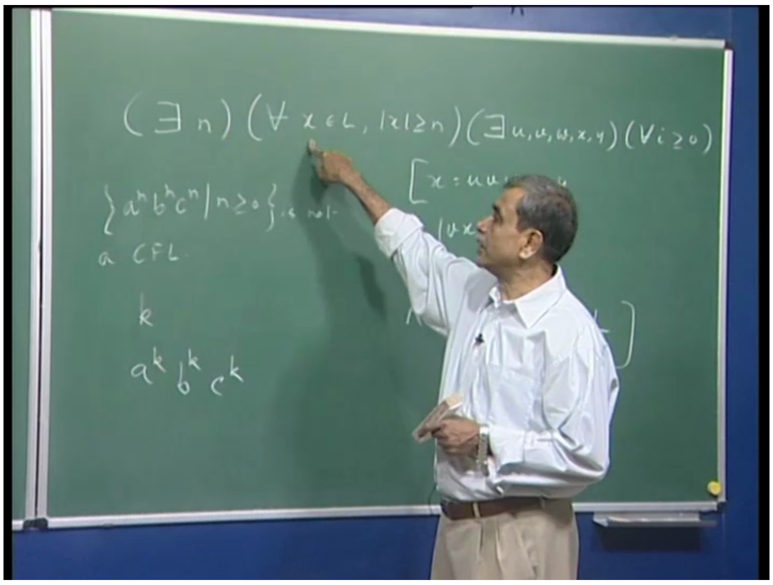
And now comes the role of adversary, the move of adversary.Adversary can breakthe string x into these five parts in any manner, right? That is not in your hand. Solet us say he breaks the string in some way, this string in u v w x y and then Ias the prover chose an i and obtain a string which is not in the language by doing this.

(Refer Slide Time: 32:37)


$(\exists n)(\forall x \in L, |x| \geq n)(\exists u,v,w,x,y)(\forall i \geq 0)$
$[x = uvwxy$
$\wedge |vx| \neq \epsilon$
$\wedge |vwx| \leq n$
$\wedge uv^iwx^iy \in L]$
$\{a^nb^nc^n \mid n \geq 0\}$ is not a CFL.
k
$a^k b^k c^k$

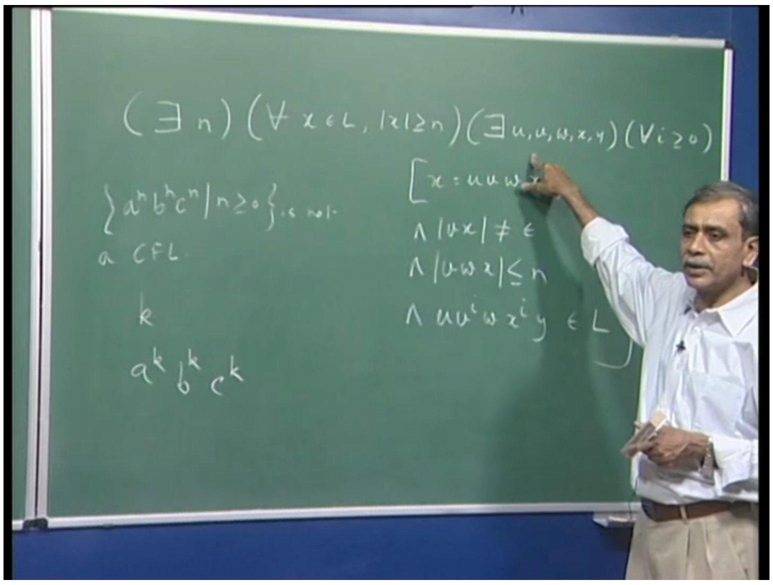
So I am showing that look we are following this game and I am getting a string which is not in the language and therefore we conclude that the language L is not contest free. So the point I wanted to emphasize was thatlook at the quantifier structure of this lemma and remember that you as the prover, you have complete freedomonly on the universal quantifier.

(Refer Slide Time: 33:12)


$(\exists n)(\forall x \in L, |x| \geq n)(\exists u,v,w,x,y)(\forall i \geq 0)$
$\{a^nb^nc^n \mid n \geq 0\}$ is not a CFL.
k
$a^k b^k c^k$

So this is something I mentioned that the time we talked of pumping lemma for regular languages. The one mistake beginners make in applying this lemma, they will say let v be this, let be this w, let this be x, let this be y and then they use some pumping and get a string out of the language.So that will not do. You should be prepared for whatever be the breakup, right? You cannot assume this breakup because this move is not in your hand, right?

(Refer Slide Time: 33:50)


$(\exists n)(\forall x \in L, |x| \geq n)(\exists u, v, w, x, y)(\forall i \geq 0)$

$\{a^n b^n c^n \mid n \geq 0\}$ is not a CFL

$[x = uvwxy$

$\wedge |vx| \neq \epsilon$

$\wedge |vwx| \leq n$

$\wedge uv^i w x^i y \in L]$

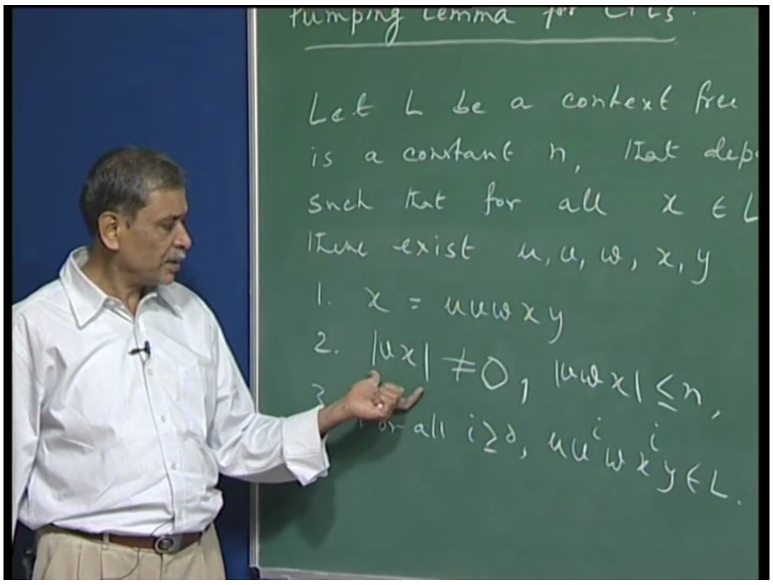
k

$a^k b^k c^k$

So now we are going to prove the lemma. I see that there is a typo you should like which is this. The way I have written here,I have been saying whenever I said that v x together is not empty but I wrote like this. But what I should have said since I am using the length v x, that what I am saying is this length v x is not equal to 0, which in other words is that both v and x, supposeeach was empty.

In that case the length would have been 0. That is not the case. So of course the main point is that at least one of v and x must be non-empty. So one way of saying that is the length of the string v x is not equal to 0, okay.

(Refer Slide Time: 34:48)


Pumping Lemma for CFLs.

Let L be a context free

is a constant n, that dep

such that for all $x \in L$

there exist u, v, w, x, y

1. $x = uvwxy$

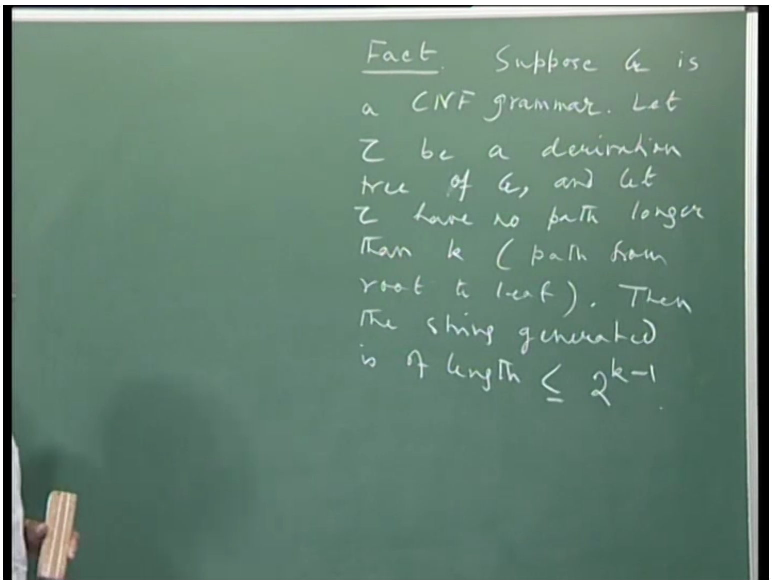
2. $|vx| \neq 0$, $|vwx| \leq n$,

3. for all $i \geq 0$, $uv^i w x^i y \in L$.

Let us go ahead and show the proof for this lemma. The one fact we are going to use is what we did in the last lecture. And so let mewrite this fact here. Suppose G is a Chomsky normal form grammar, okay. And let tau be a derivation tree in this grammar of G and let tau have no path longer than k. In other words every path, here part always means from the root to the tree.

So let me write that out, path from root to leaf, okay. So no path is longer than k then the string generated is of length less than or equal to 2 to the power k minus 1, okay.

(Refer Slide Time: 37:21)

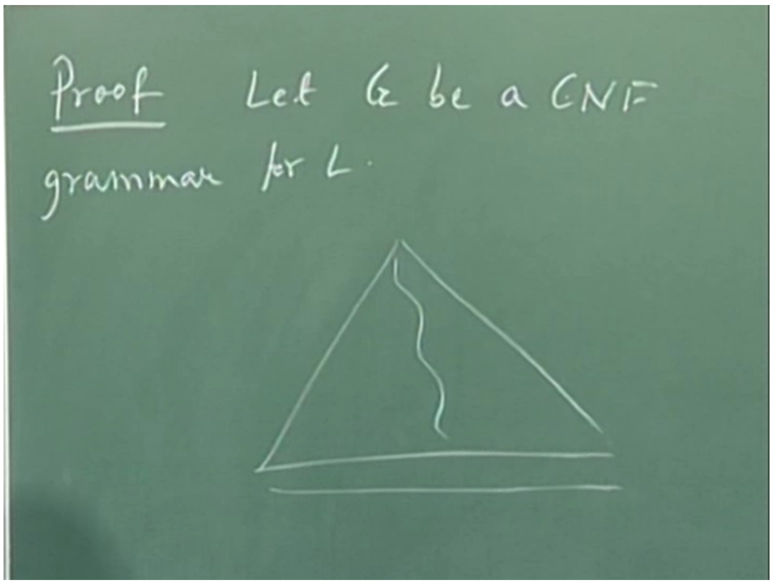


Fact. Suppose G is a CNF grammar. Let τ be a derivation tree of G, and let τ have no path longer than k (path from root to leaf). Then the string generated is of length $\leq 2^{k-1}$.

So what it is saying is that if you take any derivation tree and the longest path has a certain bound then using that bound you can bound the length of the stringthat this derivation gives rise to. So how do we use that? We say that for this pumping lemma proof we say that you know let L be a context free language. So we start with that, right? That I have in the statement of the lemma. So we will start with that let G be a Chomsky normal form grammar for L.

There is some issue about that supposeL has the string epsilon or L is empty language and so on. So these are the cases which you can dispose of. Iwill not go into that but that is not going to stop us from proving this lemma.

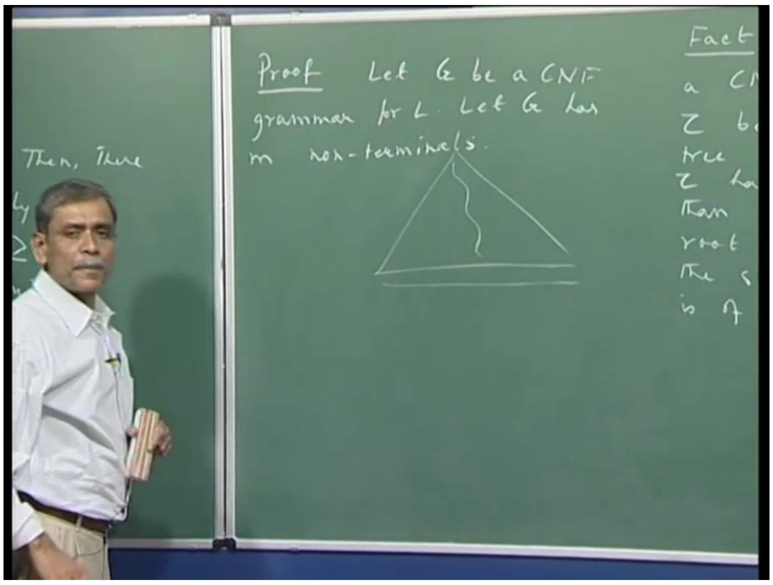
(Refer Slide Time: 38:48)



Proof Let G be a CNF grammar for L.

So remember because when I say that G is a Chomsky normal form grammar for L but then Chomsky normal form grammar exist provided L does not have an empty string. I leave it to you to argue later on that. That is not a problem if Lhas the string epsilon then you could consider the language L dash which is L from which epsilon has been removed, okay. So anywaywithout loss of generality we can consider a Chomsky normal form grammar G for the language L and let G has m nonterminals, okay.

(Refer Slide Time: 39:49)



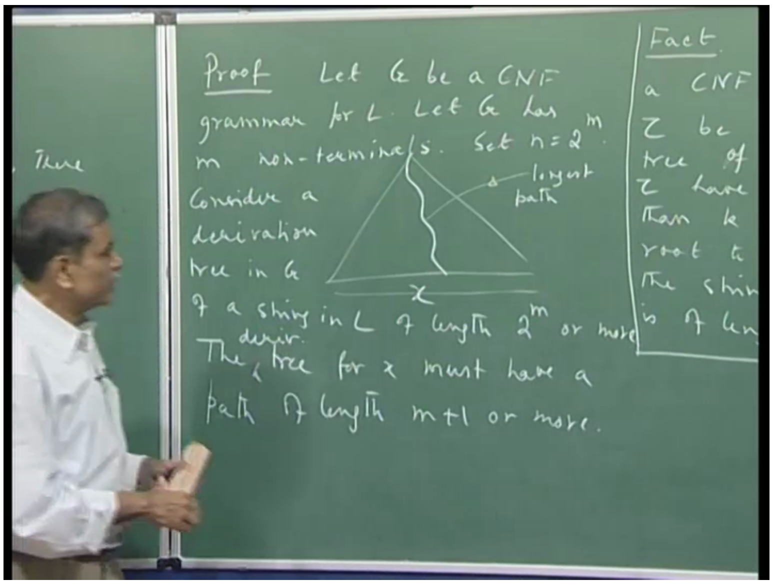
Proof Let G be a CNF grammar for L. Let G has m non-terminals.

Now you know let setn to be 2 to the power m, right? So basicallyconsider a derivation tree, right, in G of a string in L, right, of length 2 to the power m or more. Now this string is being derived, so this is the string x whose length is 2 to the power m or more and that is being derived by this derivation tree of G. What isthe length of the longest path? Do you realize that the length of the longest path in G has to be at least m plus 1? Why?

And that is why the fact comes in that suppose the length of the longest path inthis derivation tree for x was of length m or less. Sonowfor k consider m so that tree which had only every path was bounded by length k could not have generated a string larger than 2 to the power m minus 1, okay. But our string is of length 2 m.

So therefore I say that the tree or derivation tree for x must have a path of length m plus 1 or more, right? So in other wordslet considerthis be the longest path. So let us say this is the longest path in the tree.

(Refer Slide Time: 43:15)

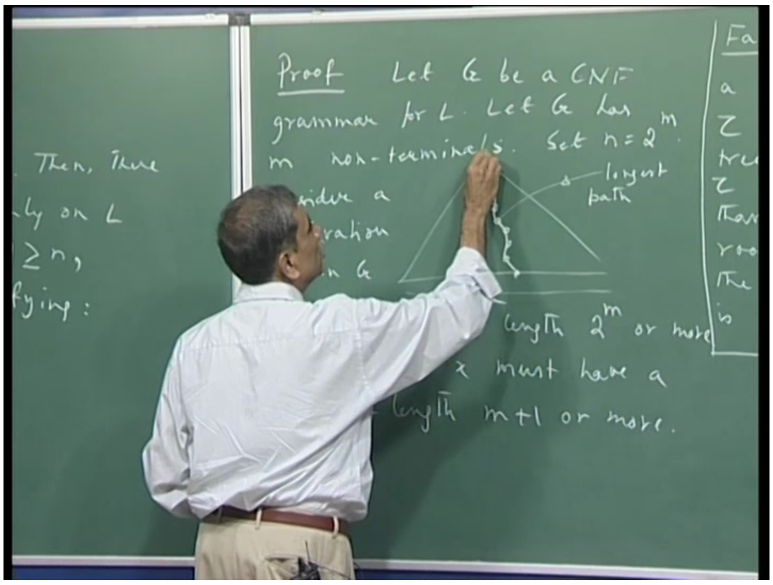


Proof Let G be a CNF grammar for L. Let G has m non-terminals. Set $n = 2^m$. Consider a derivation tree in G of a string in L of length 2^m or more.

The deriv. tree for x must have a path of length m+1 or more.

So what we are saying is the longest path therefore must be of length m plus 1 or more, okay. So now this is of course a terminal symbol andall these are nonterminal symbols, okay.

(Refer Slide Time: 43:37)


Proof Let G be a CNF grammar for L. Let G has m non-terminals. Set n = 2^m.

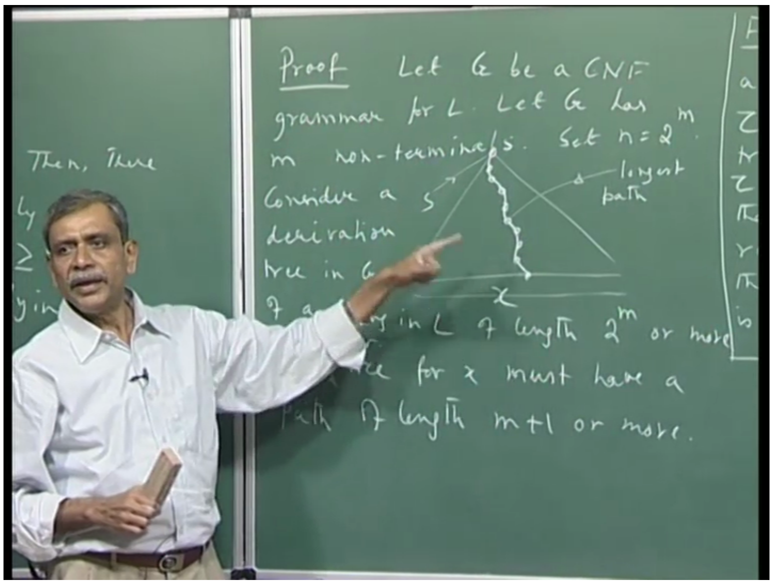
length 2^m or more

x must have a

length m+1 or more.

And this nonterminal is of course S and this path is of length at least m plus 1. So how manynonterminals occur in this path? Clearly m plus 1 or more. Sorry this path is at least of length m plus 1 so there will be at least m plus 1 nonterminals in this path.

(Refer Slide Time: 44:05)


Proof Let G be a CNF grammar for L. Let G has m non-terminals. Set n = 2^m.

Consider a derivation tree in G of a

in L of length 2^m or more

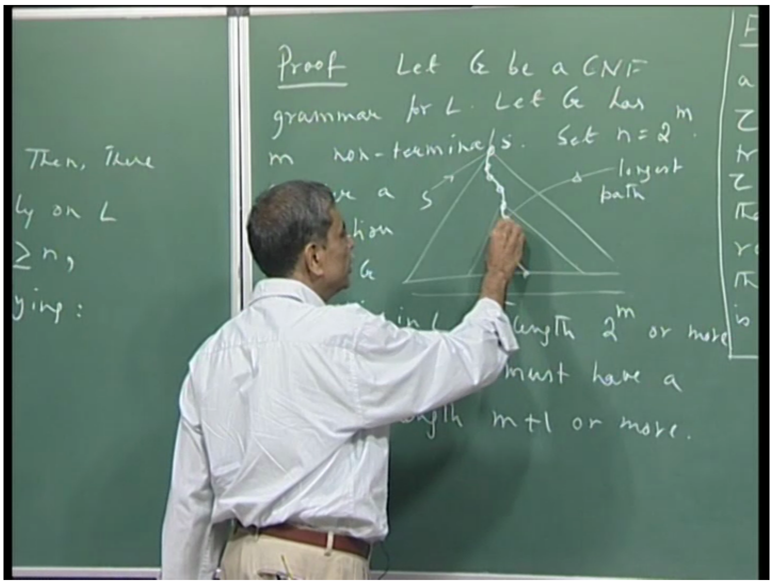
for x must have a

path of length m+1 or more.

But there are only m nonterminals in the grammar G. So if I start from hereyou knowso basically what I am trying to say is I start from this point and stop when I find some nonterminal has repeated for the first time. There has to be some nonterminal will definitely repeat in this path because that comes out of the Pigeonhole principle.

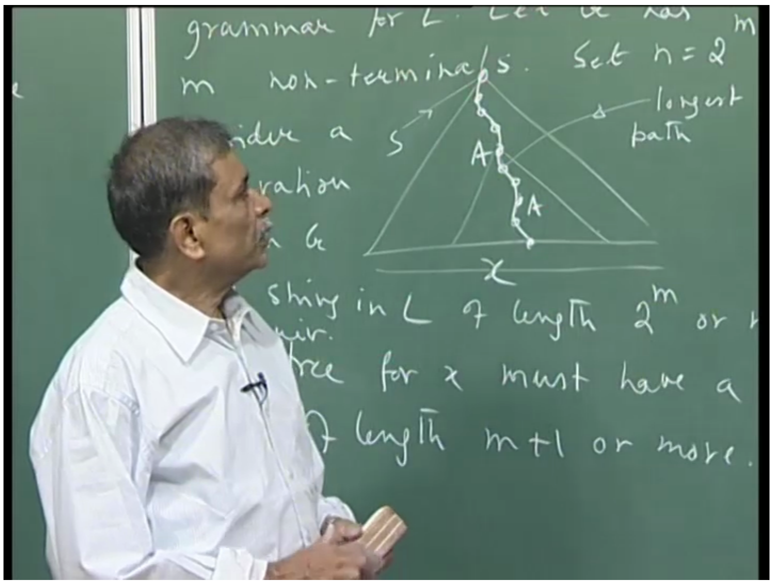
You have m plus 1 nonterminals in this path but there are only m distinct nonterminals in the grammar G. Soat least two of these must be identical. And so you are starting from here, here, here, here, here.

(Refer Slide Time: 45:03)


Proof Let G be a CNF grammar for L. Let G has m non-terminals. Set n = 2^m.
longest path
S
length 2^m or more
must have a
m+1 or more.

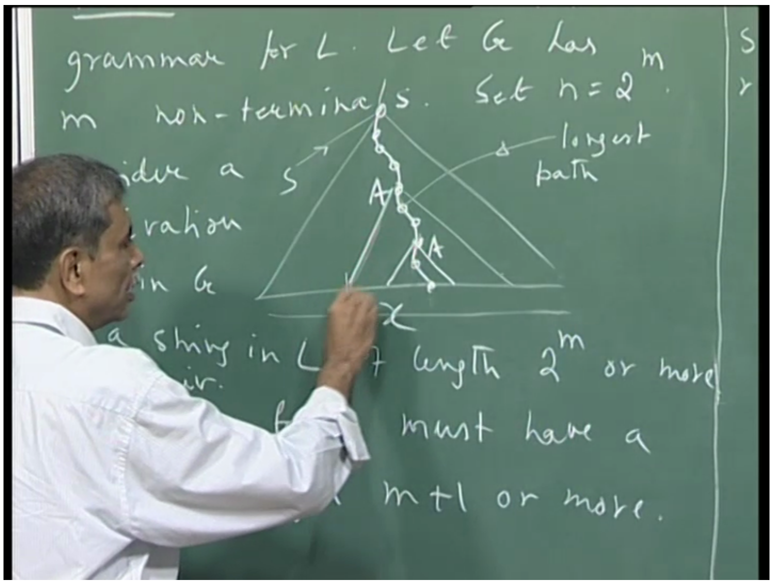
So by that time you go up,you knowyou will go up m plus 1 edges, you will certainly encounter at least one repeat. So remember the repeat is let us say the first time the nonterminal that is repeating is A, okay. And remember that this pathyou know if you are going from here, the first time A is occurring here, the next time A is occurring here. This path is of lengthat most m plus 1. It cannot be more than that. Why? Because by m plus 1 some nonterminal must repeat.

(Refer Slide Time: 45:57)


grammar for L. Let G has m non-terminals. Set n = 2^m
longest path
S
A
A
x
string in L of length 2^m or
tree for x must have a
of length m+1 or more.

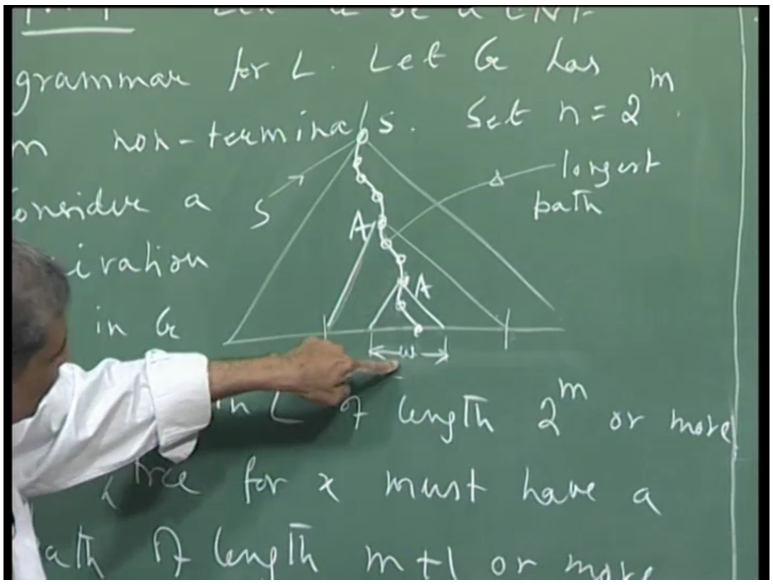
So remember that we are considering the longest path and in that longest path we start from the leaf here and you know we are going up the path till a nonterminal repeats for the first time. So this is the picture that is what I am showing, alright? So now what I would like to do is to do this. So in fact let me show you this, okay. So the sub tree subtended at the first A that generates the string let us saywhich is this part, right?

(Refer Slide Time: 46:45)


grammar for L. Let G has m non-terminals. Set $n = 2^m$.
...ider a ...ivation ...in G ...a string in L of length 2^m or more ...ir. ... must have a ... m+1 or more.

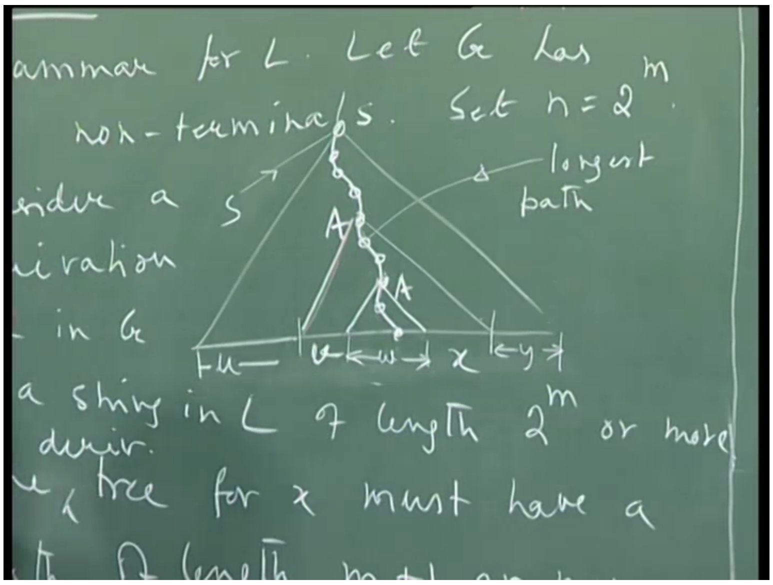
Soyou know I do not know how to put it here butyou can see this A, let this A generatethis part of the string and let this A generate this part of the string, right? So let me say that this part is called w, right? See in picture what I am trying to show that A is deriving this string w.

(Refer Slide Time: 47:27)


grammar for L. Let G has m non-terminals. Set $n = 2^m$.
Consider a ...ivation in G ... in L of length 2^m or more ... tree for x must have a ...ath of length m+1 or more

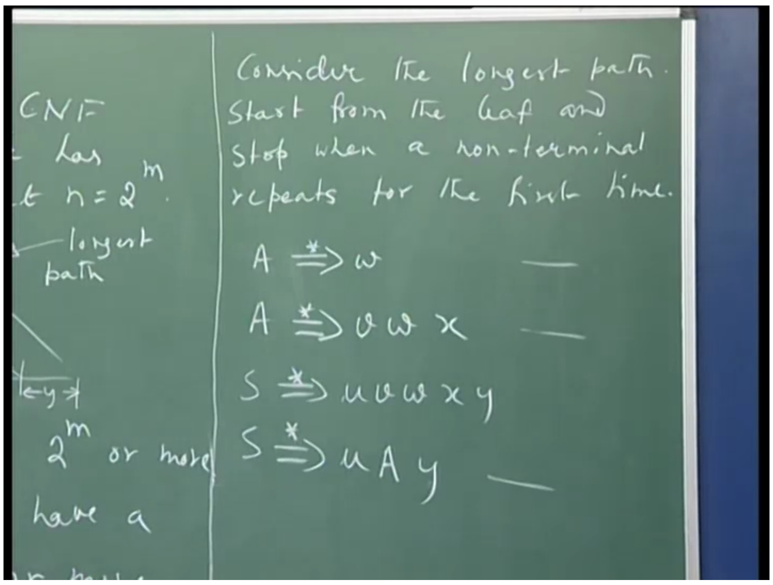
So let me write it.A derives w. And this A derives of course this w as well as something before it maybe. Let me call that v. Something right after w, x. In other words we are also saying because you see this is also A, so this is deriving v w x. And what does S derives? You can see S derives, so let me say this part is u and this part is y. So is it clear?

(Refer Slide Time: 48:30)



So S is deriving the entire string of course u v w x y and we have these two also and particular of course S is also deriving u A y, right? So just consider this. S is deriving u and thenthis A part was deriving v w x and S was deriving rest of the string y. So nowconsider this, this and this, right?

(Refer Slide Time: 49:23)


Consider the longest path. Start from the leaf and stop when a non-terminal repeats for the first time.

$A \overset{*}{\Rightarrow} w$ —

$A \overset{*}{\Rightarrow} v w x$ —

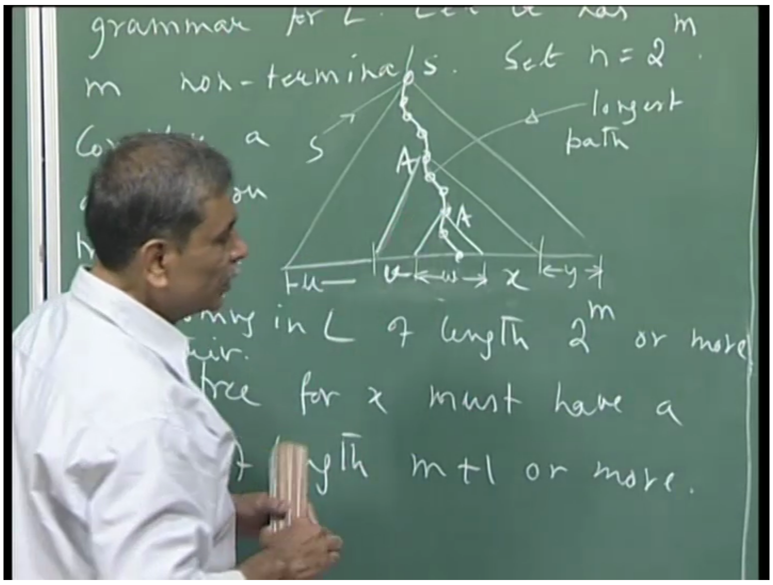
$S \overset{*}{\Rightarrow} u v w x y$

$S \overset{*}{\Rightarrow} u A y$ —

CNF
has
t n = 2^m.
longest path
y
2^m or more
have a

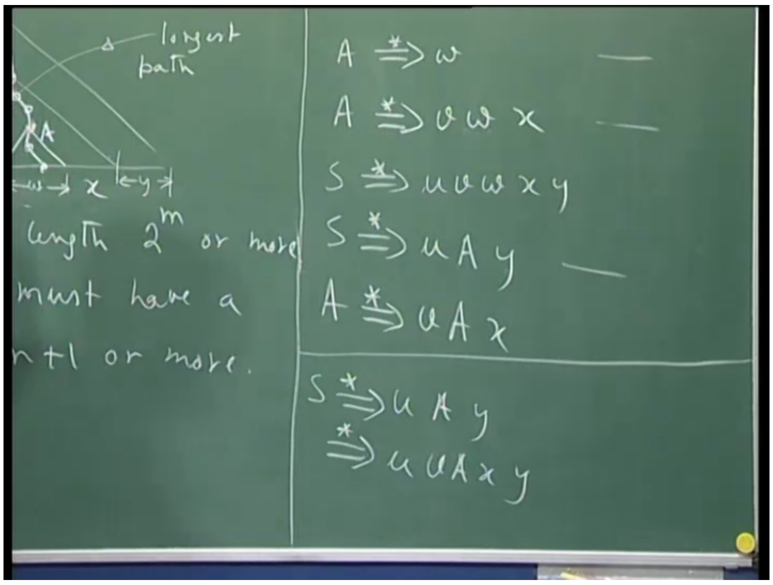
We can also say that this A was deriving v A and then x, okay. This A was deriving w, this A was deriving v w x. So this w was of course derived by this A. Remember this was deriving v w x of which the w part was derived from A.

(Refer Slide Time: 50:04)



So clearly A was deriving v A x, right? And now you can see the reason why the pumping is happening, okay. So using S you came to u A y, right? Then,so u A y, this A can be replaced of course by this v A x. This is u, this A is replaced by v A x y.

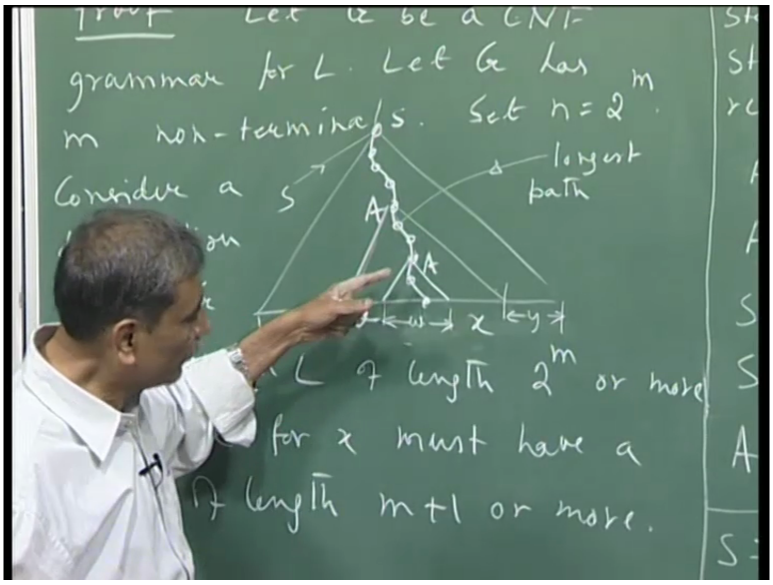
(Refer Slide Time: 50:48)



longest path
A
w x y
length 2^m or more
must have a
m+1 or more.
$A \overset{*}{\Rightarrow} w$
$A \overset{*}{\Rightarrow} v w x$
$S \overset{*}{\Rightarrow} u v w x y$
$S \overset{*}{\Rightarrow} u A y$
$A \overset{*}{\Rightarrow} v A x$
$S \overset{*}{\Rightarrow} u A y$
$\overset{*}{\Rightarrow} u v A x y$

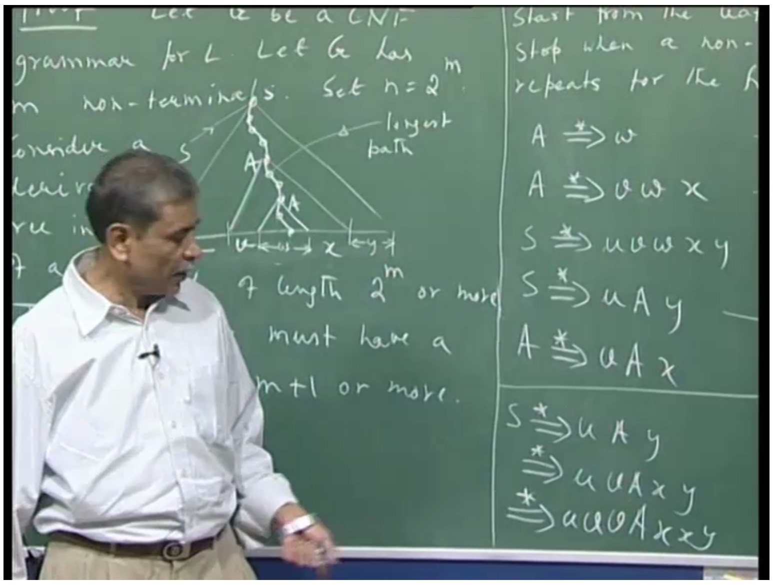
And in this string what we did was replace or rewrite this A or we derived from this A the string w.

(Refer Slide Time: 50:56)



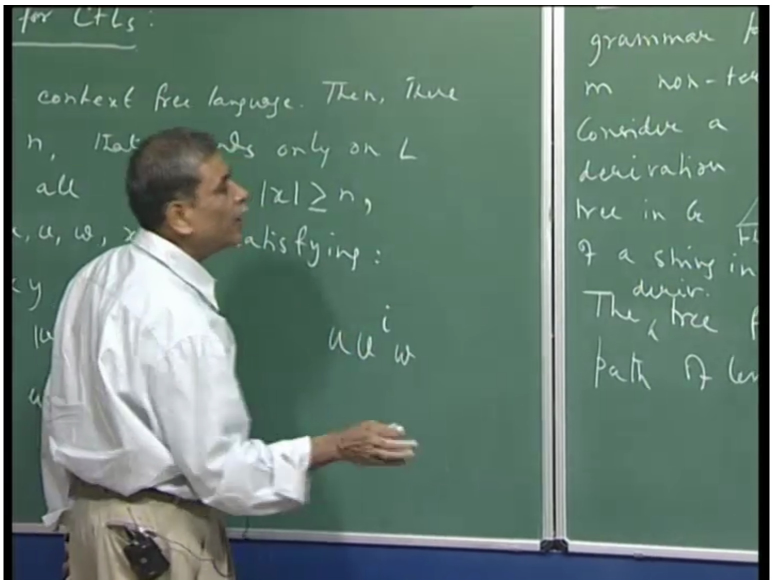
But you know since I have this nonterminal A, I can repeatedly apply this A going to this v A x. So I could do this that u v and then this A I rewrite as v A x, v a X, and then of course thesex and y would come, right? So what is the string that you have now? That S derivingu v v A x x y.

(Refer Slide Time: 51:34)


Start from the
Stop when a non-
repeats for the
$A \overset{*}{\Rightarrow} w$
$A \overset{*}{\Rightarrow} v w x$
$S \overset{*}{\Rightarrow} u v w x y$
$S \overset{*}{\Rightarrow} u A y$
$A \overset{*}{\Rightarrow} v A x$
$S \overset{*}{\Rightarrow} u A y$
$\overset{*}{\Rightarrow} u v A x y$
$\overset{*}{\Rightarrow} u v v A x x y$

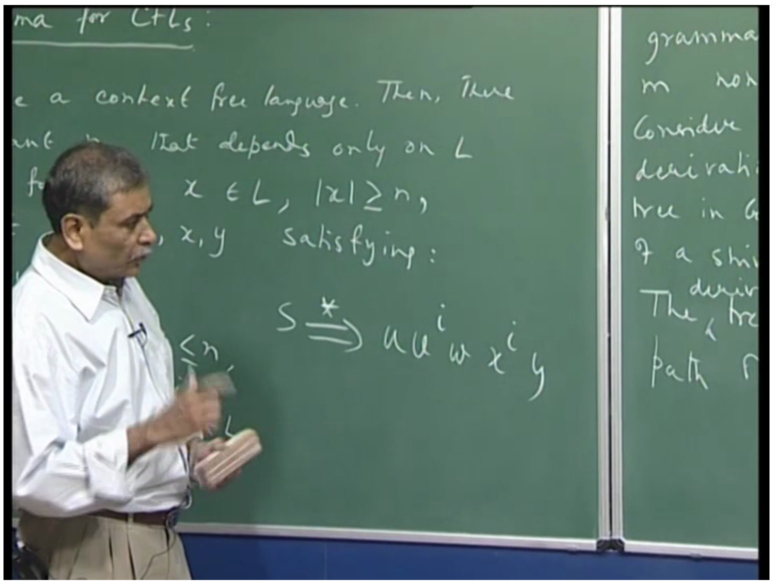
You need not stop here. You can again rewrite A as you knowv A x. So then you will get three v’s and three x’s then y and so on. So (ther) therefore you can see I can derive any string of this form u v i then w.

(Refer Slide Time: 52:09)



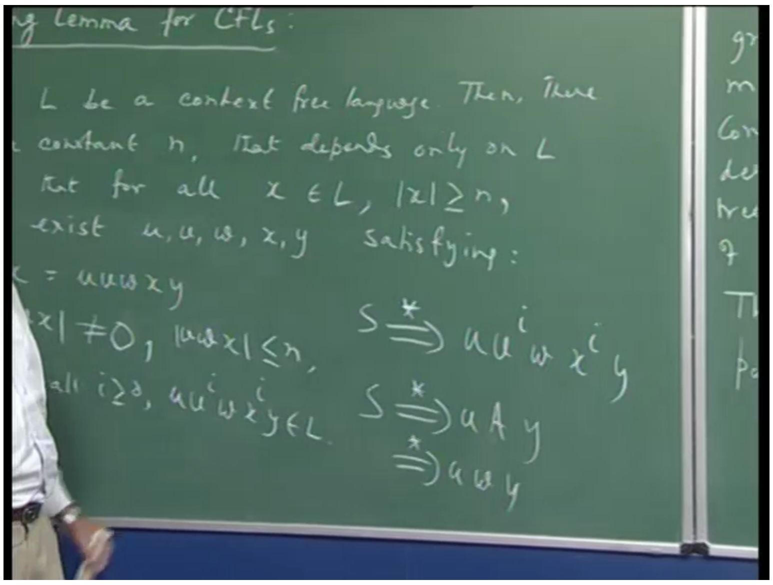
You know finally when I am tired of applying this rule again and again and again, A going to v A x, finally I rewrite this A as w, okay. SoI will obtain from S the string u v i w x i y, right?

(Refer Slide Time: 52:34)



Now of course pumping down would mean what? That from S when you generated u A y then instead ofrewriting this A as v A x,we couldsimply rewrite A or derive w from this A. So I could get from here u w y, okay.

(Refer Slide Time: 53:10)


g Lemma for CFLs:

L be a context free language. Then, there
constant n, that depends only on L
hat for all $x \in L$, $|x| \geq n$,
exist u, v, w, x, y satisfying:
$= uvwxy$
$x| \neq 0$, $|vwx| \leq n$,
all $i \geq 0$, $uv^iwx^iy \in L$.

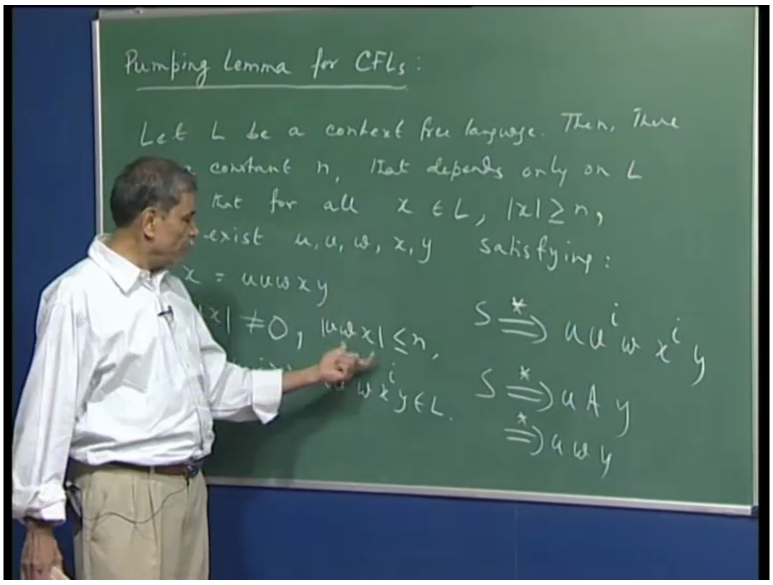
$S \xRightarrow{*} uv^iwx^iy$

$S \xRightarrow{*} uAy$
$\xRightarrow{*} uwy$

And this therefore shows that for every i greater than equal to 0, in case of 0 I get the string u w y. For other things i(equa) greater than equal to 1, then I will get u v w x y, u v v w wx x y and so on. So that is really the proof of the pumping lemma for CFLS. Now I will tell you and in fact it is not too difficult to seewhy these two conditions must apply, that v x is not empty, right? V x is not an empty string.

That means at least one of v or x must be non-empty. The other thing was that this condition,the length of v w x is less than equal to the length of the pumping lemma constant.

(Refer Slide Time: 54:07)


Pumping Lemma for CFLs:

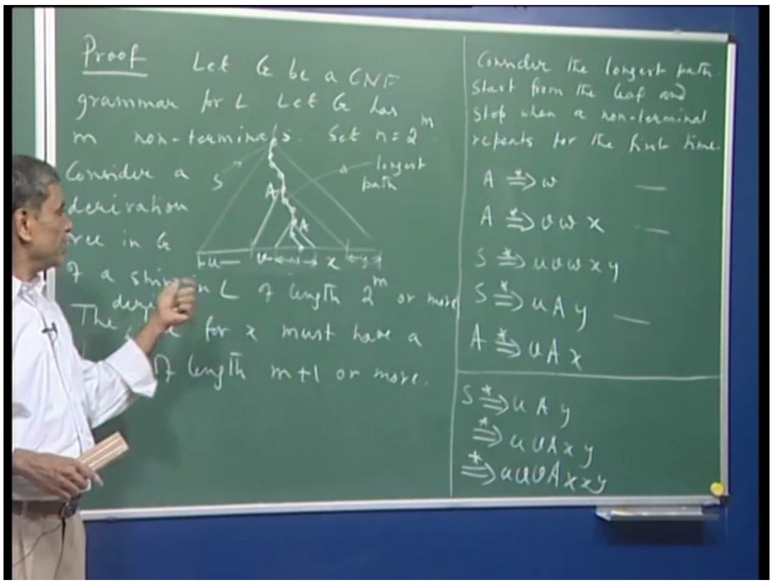
Let L be a context free language. Then, there
constant n, that depends only on L
hat for all $x \in L$, $|x| \geq n$,
exist u, v, w, x, y satisfying:
$x = uvwxy$
$x| \neq 0$, $|vwx| \leq n$,
$wx^iy \in L$.

$S \xRightarrow{*} uv^iwx^iy$

$S \xRightarrow{*} uAy$
$\xRightarrow{*} uwy$

This I will explain in the next class butactually if you just look at this picture you should be able to find for yourself the reason for these two conditions.

(Refer Slide Time: 54:22)



Proof Let G be a CNF grammar for L. Let G has m non-terminals. Set $n = 2^m$

Consider a derivation tree in G of a string in L of length 2^m or more.

The derivation for x must have a [illegible] of length m+1 or more.

S
A
A
longest path
u
v
x
y

Consider the longest path. Start from the leaf and stop when a non-terminal repeats for the first time.

$A \overset{*}{\Rightarrow} w$

$A \overset{*}{\Rightarrow} vwx$

$S \overset{*}{\Rightarrow} uvwxy$

$S \overset{*}{\Rightarrow} uAy$

$A \overset{*}{\Rightarrow} vAx$

$S \overset{*}{\Rightarrow} uAy$

$\overset{*}{\Rightarrow} uvAxy$

$\overset{*}{\Rightarrow} uvvAxxy$

We stop here.