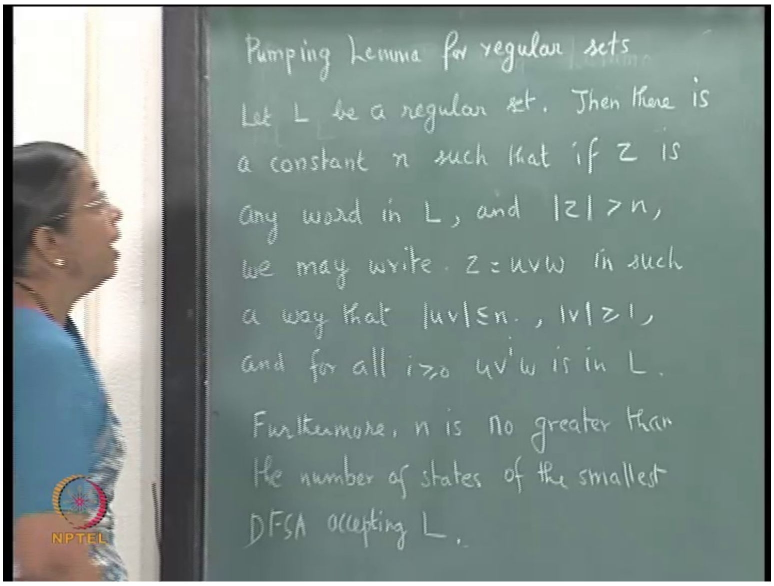
Theory of Computation

Prof.Kamala Krithivasan
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture No. #16
Pumping Lemmas For Regular Sets And CFL

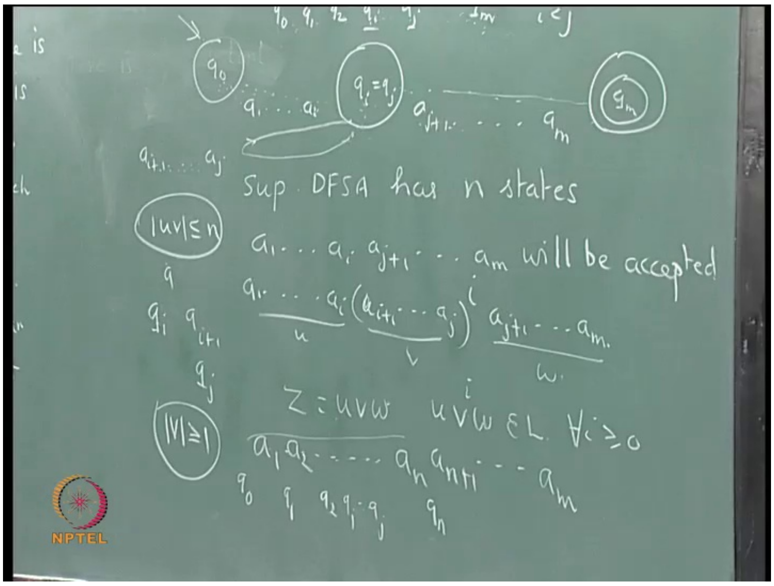
So, today we shall consider pumping lemma for regular sets. This is very useful in proving certain sets are not regular. For example, you know that a power n b power n is not regular. You cannotgenerate itwith a right linear grammar or you cannot accept it with afiveFSA. Why?The finite state automaton has a finite amount of memory,and if you want to accept a power n b power n.It should somewhere remember n, but it cannot remember infinite number of values. So, it is not possibleintuitively,this is the argument.Butformally, we can use what is known as the pumping lemma?And prove it.

(Refer Slide Time: 01:02)



Now, what is the pumping lemma? The pumping lemma is like this, let L be a regular set,then there is a constant n such that if Z is a word in L and Z,the length of Z is greater than n. Then, you can write Z in the form u v w,such a way that the length of u v alone is less than or equal to n,and v is non empty,its length is greater than or equal to 1,such that for all i greater than or equal to 0,u v power i w is in L.This is called the pumping lemma for regular sets.

(Refer Slide Time: 01:50)



The idea is very simple,look at this structure of a word Z accepted. Suppose Z is a1, a 2,a m,and m is greater than ngreater than or equal to n,actually does not matter.You start with q naughtand after reading some portion, you reach a stateq i,and thenyou reach a final statesay q m.Now, initially start with q naught,after readinga 1 you go to some key q 1, after reading a 2 you go to q 2,and soon.After reading a m you go to the final stateand a 1, a 2, a m is accepted by the automata.

Now, if m is greater than n,how many states are there? There arem plus 1state here.Suppose the DFSA has n states?supposed f DFSA hasn states, the n correspondsreally to the number of states.Furthermoren is no greater than the number of states of the smallest DFSA accepting L. So, L is accepted by amachine having n statesand Z is accepted by going from qnaught to a final state q m.

Now, you see thatq naught q 1 q 2 there are m plus 1 of them and you are having only n states. So, by pigeonhole principle, some q i and someq j will be equal.They will be the same; you cannot have m plus 1 different states. So, two of them will be equal. So, fromsay q i is equal to q j.Then starting from q naught and i is less than j say i is less than j,then starting from q naught after reading a 1 a 2a i you go to q i.

Then,after reading a i plus 1to a j,you go to the same stateq j is the same asq i. And, from a j plus 1 toa m.You go from this to the final state you have a string Z a 1 a 2 a m accepted by the machine and the length of that is greater than n the number of states of the deterministic automata.Then there is a sequence of states which leads you to acceptance andthese states all cannot be different from q naught. After reading a 1 it goes to q 1, after reading a 2 it goes to q 2 andsoon.

There are m plus 1states in this sequence and total you have only n different states. So, two of them have to beequal by pigeonhole principle. So, q i is equal to q j,say where i is less than j,then starting from q naught, after reading a 1 a 2 a i, you go to this state. And after reading a i plus 1 a i plus 2 to a j again you go to this state, and a j after from here, after reading a j plus 1 a m, you go the final state .

Now, you can see thata 1 a 2a ia j plus 1a m will be accepted.Is not it from here to here, then here to here. It will be acceptedand then a 1 to a ithen a i plus 1 to a j.Starting from here,you can go here and you can traverse this path any number of times you want.And, then go to the final state, this will also be accepted by the same DFSA. It is a very simple concept. So, Z you can write this, you can call as u and this you can call as v and this you can call as w.

So,Z, you can write asu v w, such that uv powers i wbelongs to L for igreater than or equal to 0.When i is 0, it is u w like this, when i is 1. Whatever you had earlier u v w i is 2 is (( )) u v v w and. Soon. So, the idea ishere is very simple,now look at the statement of the lemma,when the length is greater than n. It may be written in this form,such that u v less than or equal to n and v length of v greater than or equal to 1 that is v is not empty. We are considering it deterministic automata length of u v is less than or equal to n. Why, because if you seea 1a 2a n an plus,you are taking a stringlike this.

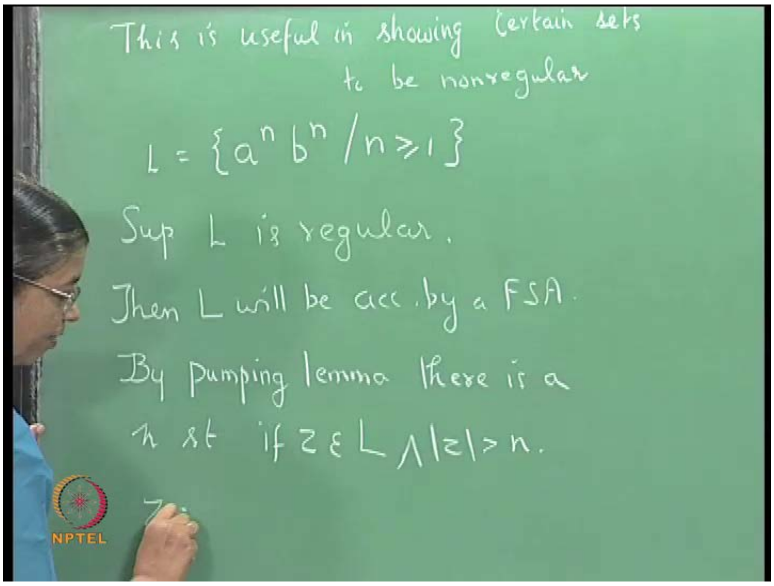
So, start from q naught,after reading a 1 you go to q 1, after reading a 2 you go to q 2after readinga n you go to q n. So, among the q naught q 1 q 2 there are n plus 1 states. Here itself something has to repeat,you have only n states in the DFSA and in this sequence you are havingn plus 1 states. So, by the pigeonhole principle here itself some q iand some q j will be equal. So, the u v portion will occur within this. So, the rest of the portion, you can consider as I mean after the pump, you can get rest of the portion as w.

So, u v portion will occur in this. So, that is why you say length of u v isless than or equal to n, length of u vis less than or equal to nand two of them are repeated even if it is a single symbol.q iq j even, if q jis q i plus 1. You would have read one symbol atleast is not ittwo of them are equal.Even, if you take theworst case that issuccessive states are identical, going from q i to q i plus 1. You would have read one symbol. So, the length of v is length of v, it is not empty, it is at least one symbol,it can be more than that. So, you have these two conditionslength of u v is less than or equal to nand length of v is greater than or equal to 1.

So, that is all the proof. So, let us look into the statement again. Let L be a regular set, then there is a constant n, such that constant is really the number of states of the DFSA.If Z is a word in L and the length of Z is greater than n, that is if you have a sufficiently long string accepted by the machine,then you can pump the middle portion u v power i Z.you can write as u v w such that u v power i belongs to L.

Note that you can also have u empty v cannot be empty, but u can be empty.The pump can occur in the beginning itself,it is possible, see from here itself the pump can occur. So, because you are putting more and more v it is called pumping lemma.You are pumping the middle portion, isnot it?vv squared v cubed like, That is why? the name pumping lemma for this

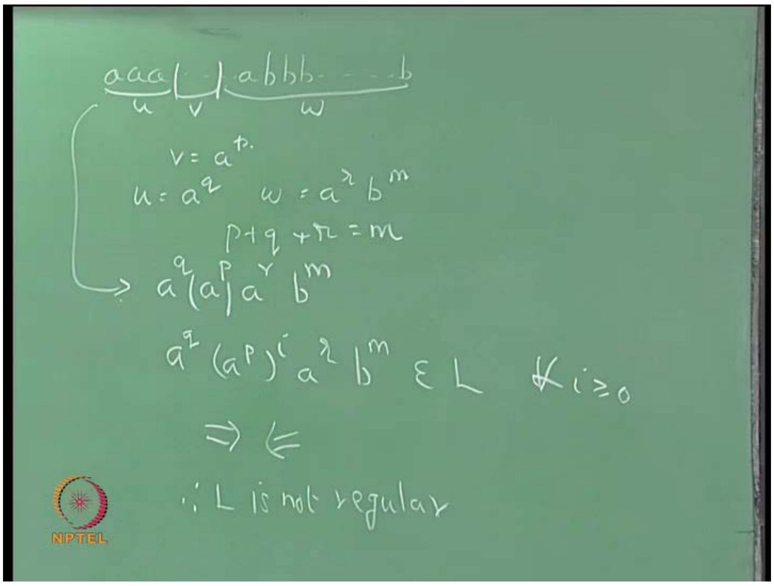
(Refer Slide Time: 11:17)



Now this is useful in showingthat certain sets are not regular.This is usefulin showingcertainsetsto benon regular;you can show some sets are not regularusing this lemma. Let us say, take the simple example a power nb power nn greater than or equal to 1.This is not regular intuitively, the argument is you cannot.Final automaton cannot remember everything if it has to accept a power n.You have it has to remember every n it is not possible with a finite amount of memory,even with the case of a carryin the case of binary adder which you consideredearlier.

It just distinguish between two classes of input histories,one is that which produce the carry and that with a whichproduce one with carry. And another without carry.It distinguish only between them, you cannot know each and every possible input history. So, if it has to accepta power n and b power n ,it has to remember after reading the entire a sequence of a. It has to remember how many a(s) it has read. That is not possible by finite state automata.This is the intuitive argumenttell us use the pumping lemmaL suppose the argument you have to write like this, suppose L isregularthen L will beacceptedby a FSAby pumping lemma.There is a nsuch thatif Z belongs to Landlength of Z greater than n.

(Refer Slide Time: 14:37)



Z can be written in the form u v wsuch that u v powers i w belongs to L. This is the pumping lemma; now considera power m b power m. Where m is greater than n. So, you have a string aaafollowed by an equal number ofb’s.Now, using pumping lemma,this can be written in the form u v w.And the other condition is length of u v is less than or equal to nand v is greater than or equal to 1. So, the pump u v portion will occur here.

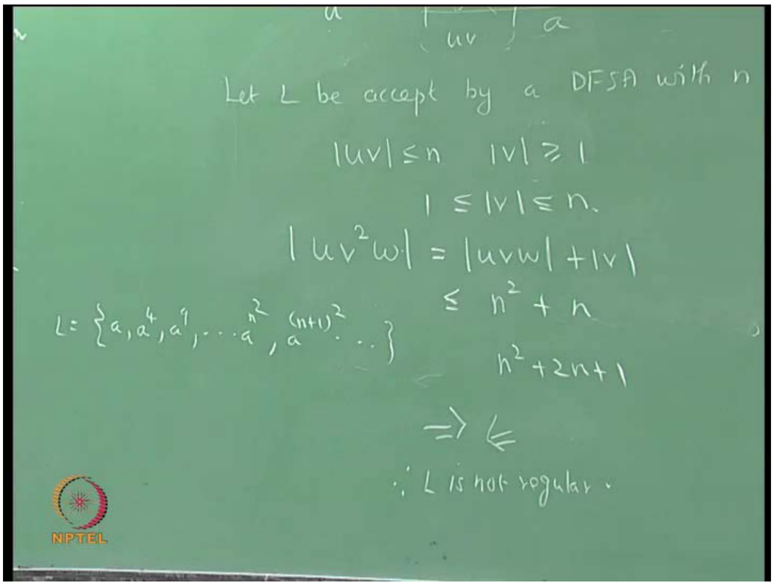
So, suppose something like this is there, this is vand this is w. We will not fall into b u v will fall within thea portion,you have chosen that way. You have taken one string aa power m b power m, where m is greater than n. So, the u v portion will occur within thissuppose v is some a powerp, then what happens u is somea power q w is a power r b power m. Where, what do you have?P plus q plus ris equal to m.Is not it. So, you have a power qa power p a power r b power m, the string is of this formp plus q plus r is equal to mand you can pump this portion.

So, you will get a power qa power pi a power rb power mbelongs to L for all i greater than or equal to 0.That means, you can get a power q a power r b power m belonging to L you can get a powera. So, manystrings you can get where the number of a(s) will not be equal to the number of b’s. If it is a regular set, it has to satisfy the pumping lemma and if it satisfies the pumping lemma, you come to the conclusion thatthere are stringsof the form a power n b power m something like that.

Where the number of a(s) is not equal to the number of b(s) and such a string will belong to the language. You come to that conclusion, but that is not correct, the language L has strings of the form, where you have only equal number of a(s) and equal number of b’s. String of a(s) followed by an equal number of b’s. So, here you are arriving at a contradiction. Therefore, L is notregular,any questions? Let me work out one more example,the argument is similar, instead of this,I will consider a powern square.

You want to show that a power n squared is not regular. Suppose L is regular, then L will be accepted by a FSA. And the pumping lemma, there is an such that if the length of n Z is greater than n. Then Z you can write in this form, such that u v power i belongs to L

(Refer Slide Time: 18:37)



Now, hereconsider a power n square,it is a string of a(s) and the first uv.Let L beacceptedbya fDFSAwith n states, that is the n is for the pumping lemma. So, if you take a power n squared, you can write it like this. And the uv portion will be willwithin the first nsymbols.

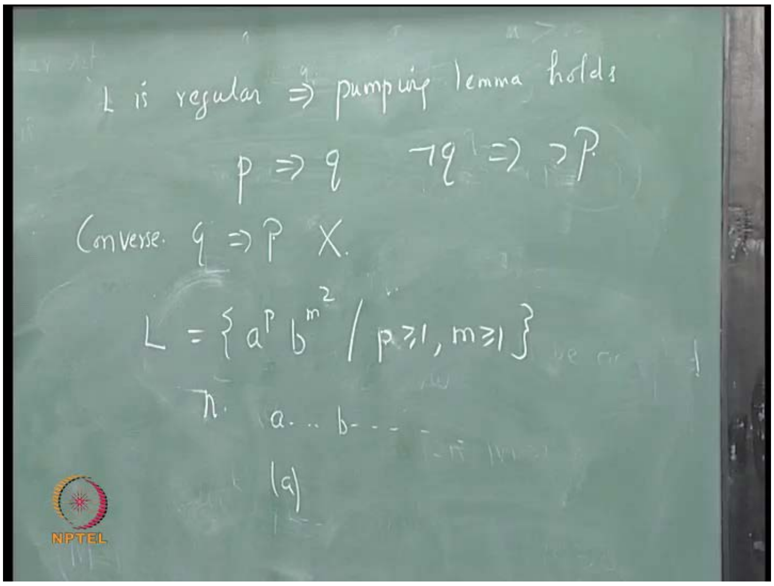
So, suppose you considerlength of uv is less than or equal to n,and length of v is greater than or equal to 1. So, length of valonewill bebetween 1 and n. And this is some portion is u some portion is v,and then it is w. So, if you consider uv squared w.What can you say about the length?The length of this will belength of uvwplus length of v. Butwhat will be the length of uv w? You have taken a power n squared. So, it will be n squaredplus whatever it is, but it is less than or equal to nis not it length of v is less than or equal to n.

So, you get something a power some p or somethingwhose value p value of p is n squared plus n.It is less than or equal to n squared plus n,it will be greater than n square. So, after n squared,the next string you get is a power n plus 1 square. Is not it?If you write a language L will be aa power 4 a power 9. Then a power n squared the next string will be a power n plus 1squaredand soon. Is not it?and n plus 1 squared is n squared plus2n plus 1.

So, by this argument you are getting a string which is between, whose length is between this and this, which is not possible.The language,the string whose length is nextlonger string than a power n squared is a power n plus 1 squared.Butby this argument, you are getting a string whose length is in between n squared and n plus 1 square which is notcorrect. So, you are arriving at acontradiction, therefore, L isnotregular.

Many other languages, you can show to benon regular using this argument.

(Refer Slide Time: 22:22)



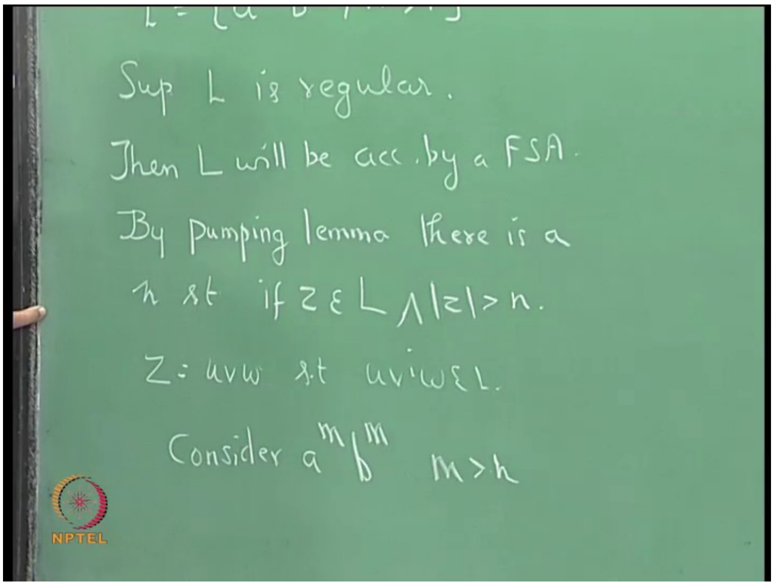
Now, pumping lemma says that L isregular,pumpinglemmaholds.The argument the way we have done is pumping lemma does not hold. So, L is not regular. So, what we are proving the(( )).

Contrapositive, we are proving the contrapositive.This is also form p implies qwhat we are showing isnaught q impliesnaught p, to show that this the argument we are giving to show that something is not regular.And it is like anyou can look at it as a sort of agame between two people.You are saying thatyou are choosing here, you want to show L is not regular.The advisorychooses n,he can choose any n, but once he chooses some n he keeps (( )). Then you chooseZ, the Z way you choose Z .You maychoose depending upon n that is what you have done.

Seeif we chose a power m b power m. Where m is greater than n or a power n squared is not it. So, the advisorychooses n and you are choosing Z,then the advisorycan write Z in the form uvwfor some uvw satisfying the conditionuv is less than or equal to nand v greater than or equal to 1.Then you show that uv power k wdoes not belong to L for some k. You show thatthis is not possible, that is why?It is not regular. This is the way it goes.

So, here if you look at the statementfor allL, L is a regular setthere exist nfor all Z.Such that Z of Z is Z belongs to L and length of Z is greater than n.There is a way to write it in the form Z is equal touvw.The statement of the pumping lemma, you look at it,then for all i Z iuv power iv belongs to L.This is the waythe pumping lemma it is sort ofyou usein alternate mannerfor any regular set,for all regular of any regular set therea nfor all Z belonging to L,such that length of Z greater thann .

(Refer Slide Time: 24:03)



z can be written in the form uvw, there is a way to write Z in the form,such that uv power i w belongs to L.And you are corresponding to thefor all quantifier, the adversary is thatthere exist quantifier. So, you first choose L then the advisorychooses n then you choose Z.Depending on n, you can also fix the a Z once you know what is n?Then adversary can write it in the form uvw. So, you show thatfor some k,uv power k does not belong to L.This is the sort of argument;we give for proving some sets are not regular.

Now, p implies q this is of the form, and what we have used is?Naught q impliesnaught p, contrapositive.What is a converse?What is a converse to pumping lemma? It will be of the form q inverse p,is not it?Converse is this, if you make a statement, it will be like this.If there is ansuch that for any Z belonging to L and L is some language,you start with L is some language. Then if there isa n such that for any Z of length greater than n.You can write Z in the form uvw such thatuv power i w belongs to L for all ithen L is a regular set.Is not it? The converse will be of that form. Is it true?Is the converse true? Do you think it will be true?

How many of you think it is true?Converse will be true? How many of you think it is not true?How many of you think that converse will be true?Youcannot say converse is not true.For that you consider that is the pumping lemma may hold L may not be regular.If the pumping lemma does not hold L is not regular, but if the pumping lemma holds you cannot conclude L is regular L may be regular or non regular you cannot say. So, take L to bea power n b powerm squaredn greater than or equal to 1m greater than or equal to 1.Consider thisis a regular set.This is not a regular setnumber, you can have a sequence of a's, but the sequence of b(s) is m squared.

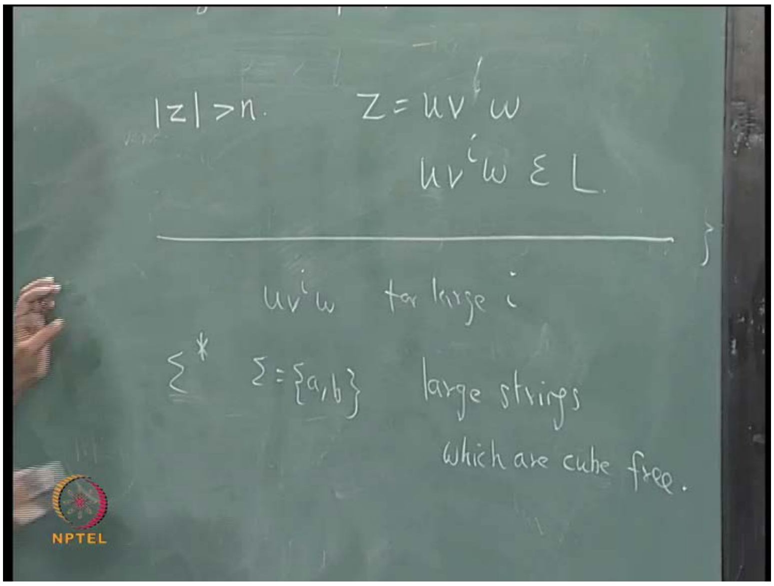
So, this is not a regular set, but if I choose ninstead of n. See Ido not want to use a same n, may be I will saya powerp,p greater than or equal to 1 and m greater than or equal to 1.Suppose this satisfies the conditionfor all there is a n,then you can choose a stringsee the p is greater than or equal to 1 there will be at least one a. So, the string will be of the form ab.

If you take a string of length n, whatever it is, the first symbol has to be a. It cannot begin with a b and the pump you can just pump athe first symbol alone you can pump.Then the resultantstring will be in the language. Is not it?Thisunionyou have to also have b power q q greater than or equal to 1.Some sequence of a(s) non empty string of a(s) followed by a sequence of b's,sequence of b(s) will be of the form m squared otherwise just sequence of b(s) alone.The reason for that,if it is if you take i to be 0and you start with the string sayabbb something. If you pump a i is equal to 0you will just get a string of b's.

This you can pump,i times i, if you take to be 0 you have to get a string of b's.Sequence of (( ))that is why thisisregular,anyway this is not regular. You can see thatthis is not regularbecause of this portionyou have a sequence of a(s) followed by the sequence of b(s) which is not of aform, it is of this form. So, it not regular, but you can have any sequence of this without a's. You can have any sequence of it this portion is regular does not matter, but the whole thing is not regular.

So, the converse to pumping lemmais not true, there is something like a converse to a pumping lemma which is a very involved theorem.It is not given in any book i think it is only in the paper. There is a result byehrenfeucht rozenberg and Parikhehrenfeucht-ehrenfeucht Parikh and rozenberg which is something like a converse to pumping lemma. Which assumes more condition, and then you show that under these conditions,it is not only pumping lemma.You identify certain positions and say that certain things can happen, occur and soon.

(Refer Slide Time: 33:05)

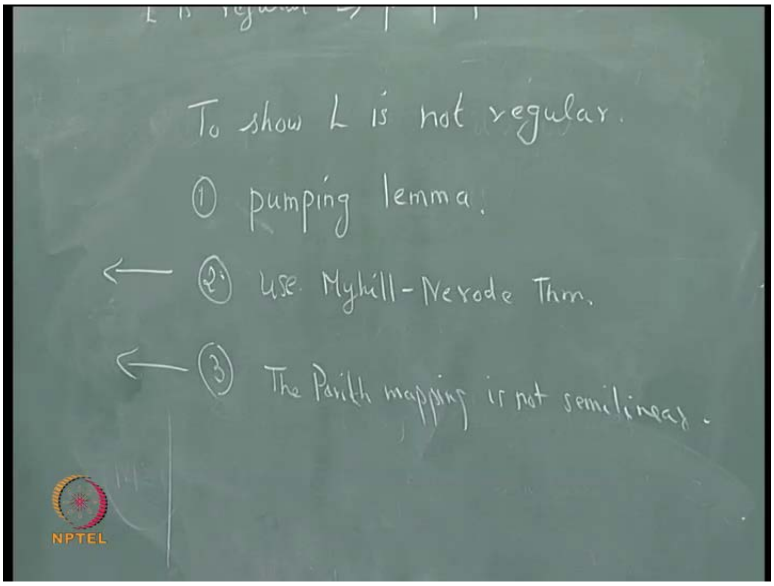


Then L will be a regular setsomething like that with this condition alone,if the converse is not true, moreoveryou must realise that. What pumping lemma says is? if you have a string,whose length is sufficiently large.Then Z can be written in the form u v power w u v wsuch that u v power iw belongs toL. You can keep on incrementing i, ican be one two three, it can go to upto infinity.

So, you will get an infinite number of stringsbeginning infinite number of strings which belong to L.It does it mean that if you have sufficiently large string belonging to L.You can write it in the formu v power i w forlarge i.That is not true, what pumping lemma says is if you have a fairly largestring, then you can write it in this form. And you can get infinite number of strings of this formit does not mean thatif you have a very large string you can write it in this form u v power i for large i.

One example is simple example is sigma star,where sigma is a b.You know that there are large strings which are cubed.We have seen that there are large stringswhich arecube,that is no sub string will occur three times consecutively. So, this is what it meansyou can make use of this to show thatother things like that w c w r is not regular a power p p is a prime not regulara power two power nn greater than or equal to zero not regular and soon.

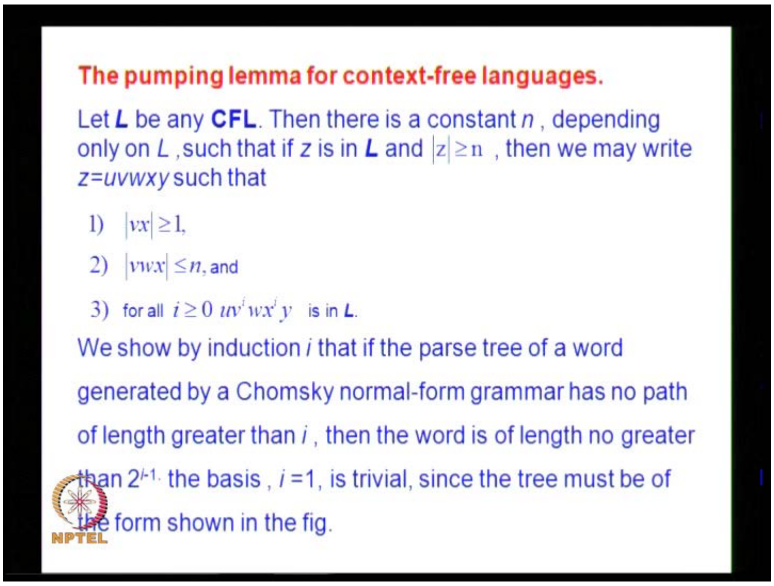
(Refer Slide Time: 35:43)



Actually, it is important to show that certain sets are not regular and there are three ways of doing that. One is use pumping lemma to show something in thatto show L is not regular.You can use three methods, one is use pumping lemma.The next method is usemyhill-nerodetheorem, this we shall be considering next.Thirdthe parith mappingis not semilinear;this is another way you can show.These twothings we shall considerlater.

Now, we shall consider the pumping lemma for context free languages,for regular sets what we found is? If the string is sufficiently long then a portion of it can be pumped any number of times. For contextfree languages, what wedefined is?Given a contextfree language, if you have a string which is sufficiently long,then two portions of the string can be pumped equal number of times. So, that we get an infinite number of strings.

(Refer Slide Time: 37:16)

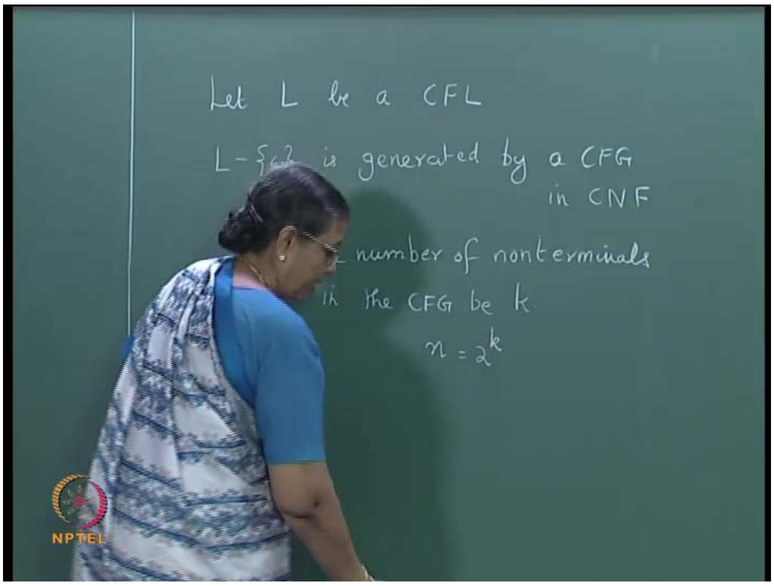


So, the statement can be like this, let L be any contextfree language.Then there is a constant n depending only on L such that if Z is in L and the length of Z is greater or equal to n. Then we may write Z as u v w x y such that length of v x is greater than or equal to one. What it means is both v and x cannot be simultaneously epsilon and second condition is length of v w x is less than or equal to n.And for all i greater than or equal to zero u v power i w x power i y isin L that is both the portions v and x can be simultaneously find equal number of times.

Now, you can see that, if you have a sufficiently long string in a language the corresponding parse tree should be,should have a path which is sufficiently long. In fact, we can show this we can show by induction on i that is the parse tree of a word generated by a chomsky normal form grammar has no path of length greater than i.Then the word is of length no greater than two i minus 1.So, in order to prove the pumping lemmafirst we shall take the contextfree grammar to be in chomsky normal form.

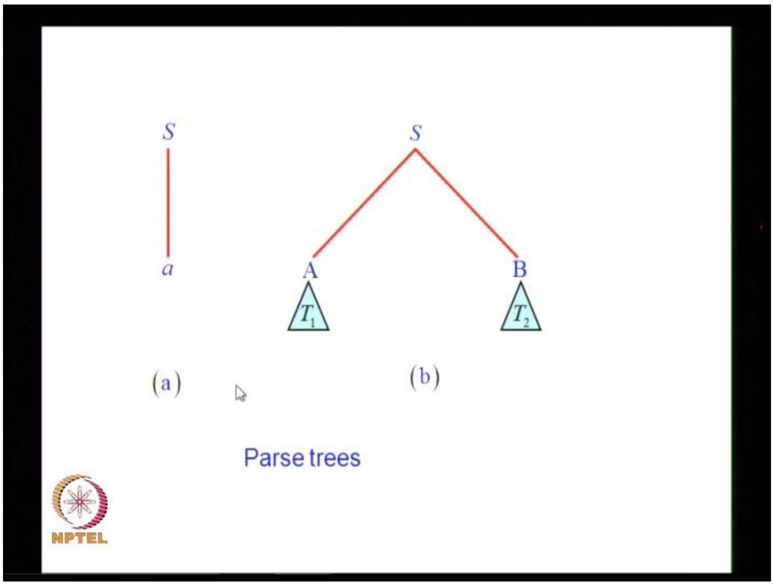
(No audio from 38:57 to 39:21)

(Refer Slide Time: 38:57)



Let the contextfree language minus epsilon, we are not very much bother about epsilon, because we are going to consider only strings of length greater than or equal to n for alarge n. So, let L minus epsilon be generated by a contextfree language in chomsky normal form.Letthe number ofnon terminalsin the c f gbe k,and then we take n to be two power k.Now in this grammar we can show thatif the parse tree of a word generated by the chomsky normal form grammar has no path of length greater than i,then the word is of length no greater than two power i minus 1,for the basis i one is trivial since the tree must be of the form shown in the figure.

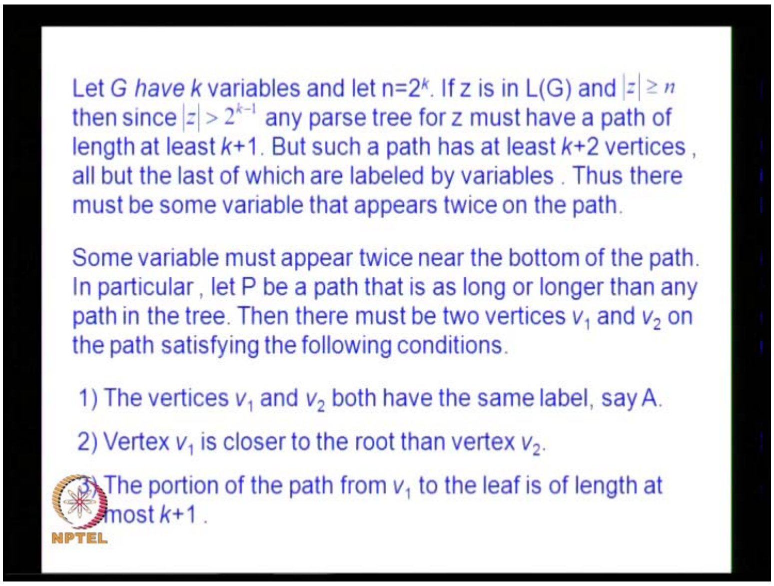
(Refer Slide Time: 40:32)



If the path length is one, you have a situation like this the length of the string generated is one you see that i is 1. So, two power one minus 1 is two powerzero which is one. So, the bases class holds.

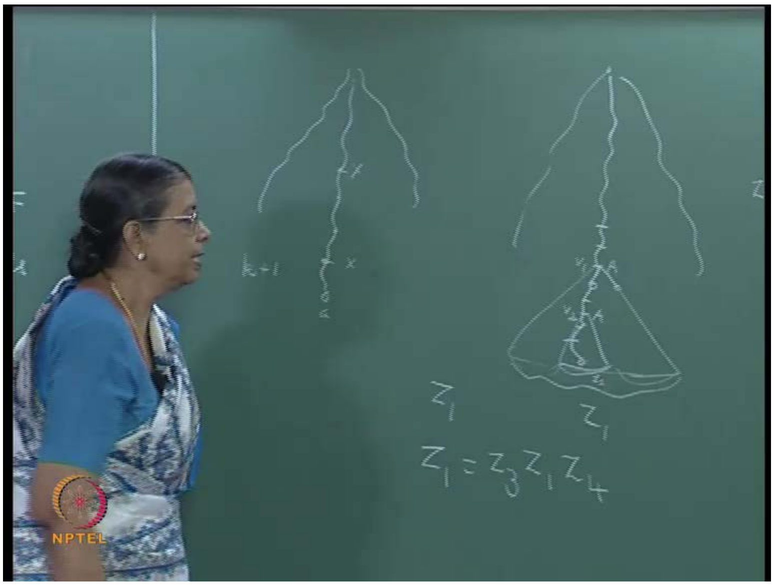
Now, for a induction assume that the result is true up to path length i minus 1 and show that the result is true for path length i. Now you consider thistree with path length maximum i, then the first step of the tree will be like this. And from thisnon terminal there will be a sub tree here.The maximum path length in t one or two t two will be i minus 1.So, the maximum length of the string generated in this portion will be two power i minus 2.Similarly the maximum length of the string generated here is two power i minus 2.So, the total length of the string generated by this tree will be two power i minus 2 plus 2 power i minus 2 which is two power i minus 1.So, this proves the simple result.

(Refer Slide Time: 41:45)



Now, let G have k variables or k non terminals and we take n to be two power kif Z is in L g and the length of is length of Z is greater than or equal to n.Thenbecause the length of the string is greater than or equal to two power k minus 1 any parse tree for Z must have a path of length atleast k plus 1.If it is less than that then the string will not be of this length, but such a path will have a at least k plus 2 vertices. And the last one will be a leaf rest of them will be non terminals and there must be some variable that will appear in the path.

(Refer Slide Time: 42:36)



So, let us consider like this,the tree hasa pathof lengthk plus 1 or more.Now, let this be the longest path there may be one or two such paths, let us take one of them.You find that the last one will be aterminal symbol rest of the nodes in this path will benon terminals.And there are only k non terminals and there are k plus 1 nodes with labels which are non terminals. By the pigeonhole principle at least onenon terminal will be repeated that is somewhere here, somewhere here,the same non terminal will be repeated as the label of the internal node.

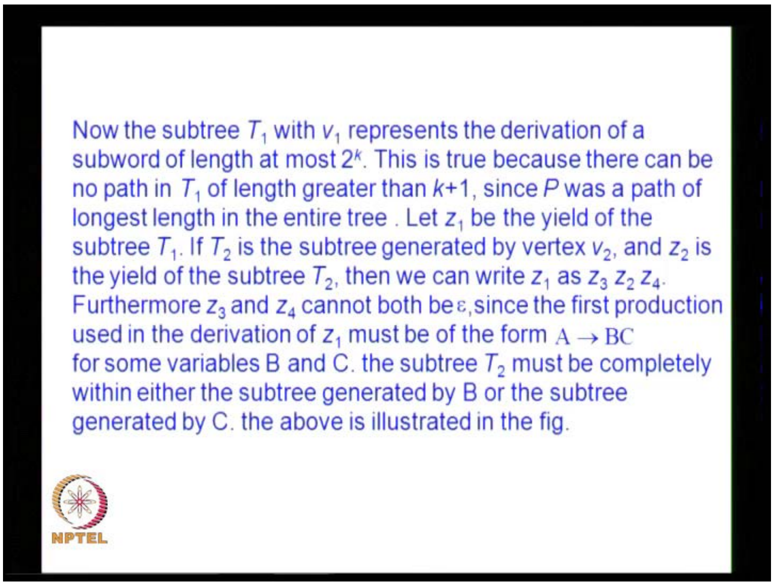
In fact, you can say something more than that;some variable must appear twice near the bottom of the path. This you can say, in fact, let p be a path that is as long or longer than any path in the tree. Then the there must be two vertices v 1 and v 2 on the path satisfying the following conditionsthe vertices v 1 and v 2.The vertices v 1 and v 2 both have the same label a, the vertex v 1 is closer to theroot than vertex v 2.The portion of the path from v 1 to the leaf is of length at most k plus 1.How can you get this?

So, in the derivation tree path Z takethe longest path,if there are more than one take one of themstart from the leaf go up the pathup to a path length of k plus 1.You consider a path length of k plus 1 here.In that there will be k plus 1 non terminal nodes that is nodes with labels as non terminals. The last one will be a leafamong thisk plus 1 by the pigeonhole principle two of them will have the same label k, saycall this as v 1,callthis as v 2.

So, v 1 is nearer to the root than v 2 and both of them have the same label. The path from v 1 to the leaf is at most k plus 1,in the worst case this can be a. Otherwise somewhere a will be here. So, the path length from v 1 to the leaf will be at most k plus 1,now if you consider this result of this tree,the string derivedlet it be Z 1.This one is Z 1and from thisanother tree will be sub tree will bederived, call thisas Z 2. So, you can write Z 1as Z 3 Z 1Z 4,this portion is Z 3,this portion is Z 4.

You can see that both Z 3 and Z 4 cannot be epsilon; simultaneously one can be epsilon possible.Butboth of them cannot be epsilon, the reason is from here,you use a Chomsky normal form grammar. So, it can be like this and this sub tree will entirely lie within one of them.Either in this case, it lies on the left of it left sub tree, sometimes it may lie on theright sub tree, but whatever it is the other tree will be non null, and so, you have this feature.

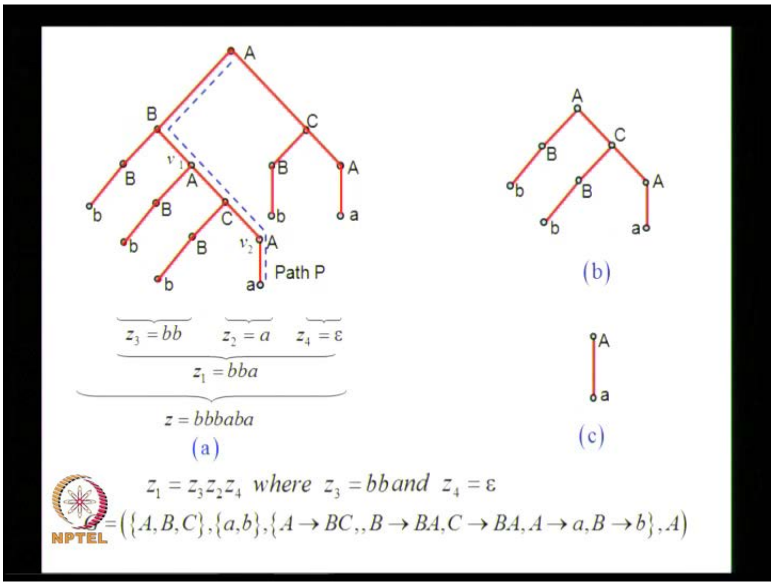
(Refer Slide Time: 46:50)



Thus you find that a tree T1 with v 1 represents the derivation of a sub word of length at most two power k.Andthis is true, because there can be no path in T 1 of length greater than k plus 1.So, in the figure what we saw is the length of the string derived from the first a,that is v 1 is at most two power k, that is why we have the first condition in thetheorem or may be the second condition in the theorem length of v w x is less than or equal to n,and p has the path of longest length in the entire tree.

Let Z 1 be the yield of the sub tree t one t two is the sub tree generated with a vertex v 2.Andthe string generated by that is Z two. So, you can write Z 1 as Z 3 Z 2 Z 4 both Z 3 and Z 4 cannot be epsilon.Since the first production used in the derivation must be of the form a goes to b cfor some variables b and c.And the sub tree t two must be completely within either the sub tree generated by b or the sub tree generated by c.

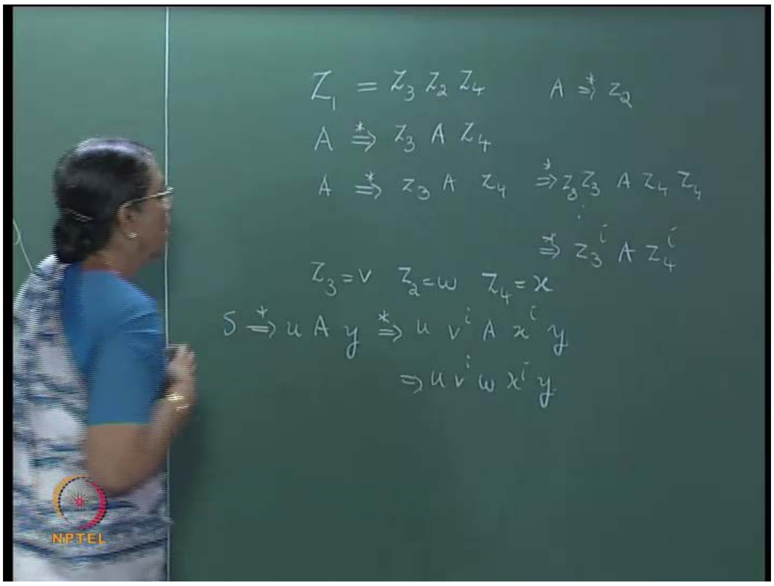
(Refer Slide Time: 48:13)



So, we have this result as an example you can consider this,you consider this example where the grammar is given by three non terminals,it is in Chomsky normal formrules are a goes to B C B goes to B A C goes to B A and Soon.a goes a b goes to b, you see that you have generation for this tree.

Now, if you consider a path, what is k here?k is three. So, consider a path of length four, start from this one two three four in this. You find that the non terminal a is repeated here. So, v 1 is nearer to the root than v 2 and v 1 v 2 have the same labelsthe result of the sub tree with this as the root. You look at this,then this generates Z 1 bba is Z 1what is Z 2 generated by this sub tree? That is just this a. So, is that one is b b a Z 2 is a Z 3 is b b this portion and Z 4 is epsilon because this entirely falls on one side.

(Refer Slide Time: 50:51)

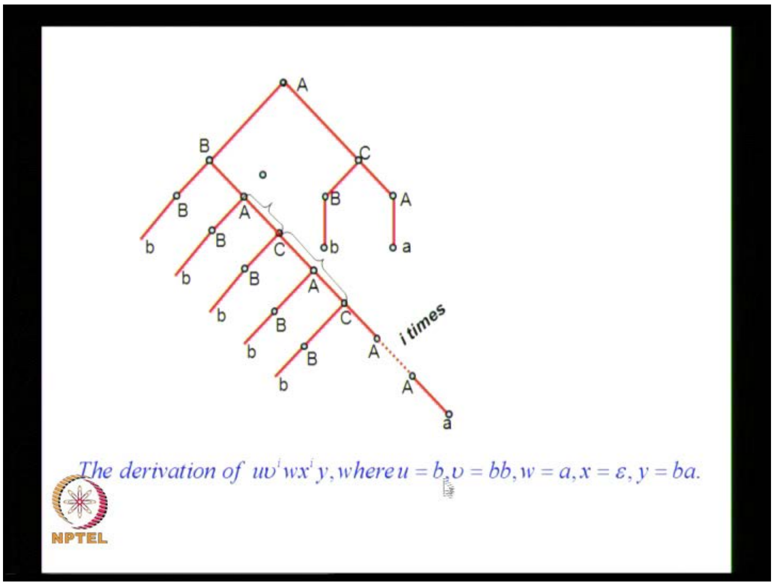


So, what you have is? Z 1 can be written in the form Z 3 Z 2 Z 4 where Z 3 is b b and Z 4 is epsilon. So, what you find isfrom the non terminal ayou are deriving Z 3 a Z 4.Now this portion you can repeat any number of times actually if you repeat this any number of times Z 3 a Z 4 Z 3 a Z 3 Z 4 Z four and soon. You can pump Z 3 anyi times and the same number of times Z 4 also will be pumped.

Now, ifin the theorem we take Z 3 to bev Z 2 to bew Z 4 to be x. Starting from syou will have a derivation ua y,if you look into this tree,some portion will be generated here, that is u. And some portion will be generated here, that is ywe are bothered about this portion. And herethis can be these two portions can be pumped.So, from this ayou can generate Z 3 and Z 4 any number of times simultaneously.That is v can be generated any number of times, x can be generated any number of times.And then finally, from a you will derive Z 2 a also derives Z 2. So, and Z 2 is nothing, but w. So, u v power i wx power i y. So, we get the pumping lemma for context free languages.

Let us see, how the pump occurs? You see thatin this example which we consider L here, see look at this example,this is Z 1. So, on the left you are getting a portion of the string which is bon the right you are gettinga string b a .

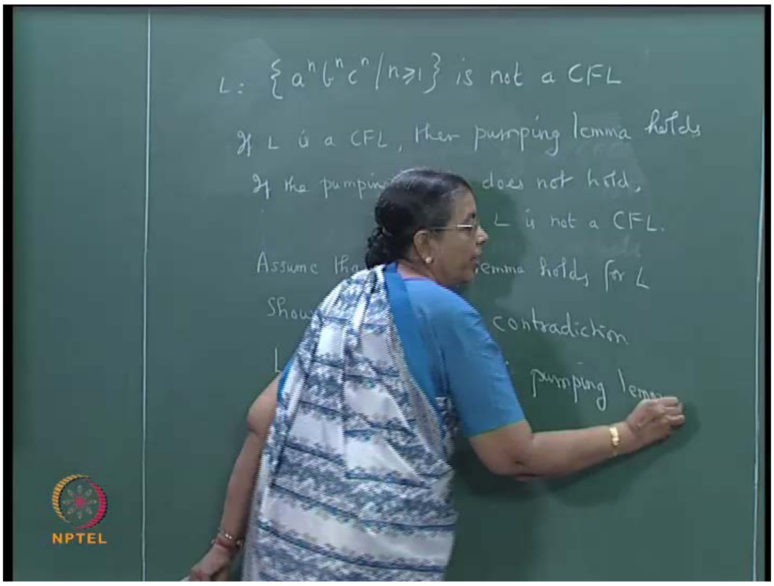
(Refer Slide Time: 53:06)



So, u is b and y is b a. So, u is band y is b aand in between this b b is the portion b.This can be pumped any number of timesb bbbbb, like that it can be pumped any number of times.x is epsilon,even if you pump it several times you are going to get epsilon only and finally.The derivation ends with the a w isa, you get here. So, this b b can be pumped any this portion is pumped any number of timesthat is what it is shown here.

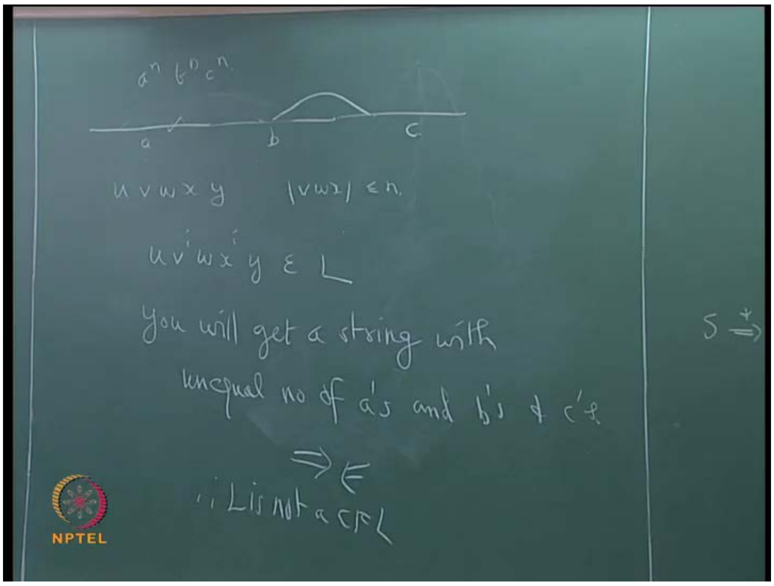
So, once twicethrice like that it can be shownit can be pumped any number of times. So, this is called pumping lemma for context free languages.How can this be used?The pumping lemma can be used to show that certain languages are not contextfree.

(Refer Slide Time: 54:13)



For example, we can show that a power n b power n c power nn greater than or equal to one isnot a CFL,it is a context sensitive language.How can we show that this is not a CFL? Actually what we are going to prove isif L is a CFL thenpumping lemma holds.The contra positive ofit will be,if the pumping lemmadoes not hold L is not a CFL. So, in order to show that,assumethatpumping lemma holdsfor L,this L then showyou arrive at a contradiction.

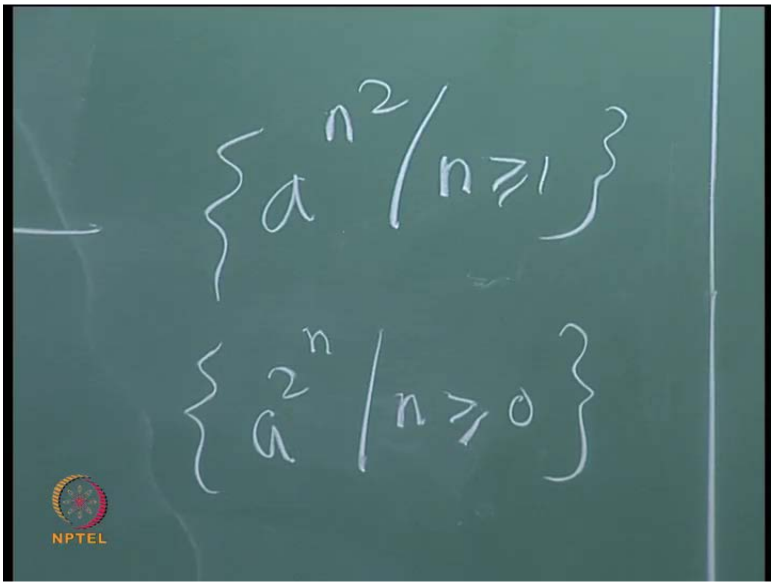
(Refer Slide Time: 56:27)



Let n be theconstant of the pumping lemma,then consider the string a power n b power nc power n. So, you have a string of a(s) equal number of b(s) equal number of c(s). Now, this can be written asu v w x y.Where, the length of v w x is less than or equal ton. So, this v w x portion can occur within a or a b or within c or like this.It cannot occur like this, it cannot include both a and c, it can include aand bor it can include b and c.It can include just one symbol that is also possible, but it cannot include or a and c.

So, when you pump u v power iw x power i y will belong to L and you will find thatwhen you pump. If the portion is like this, v w x portion is like this, the number of a(s) will increase number of b(s) will increase. But the number of c(s) will remain the sameif the portion islike this. Number of b(s) and c(s) will increase, but the number of a(s) will remain the same. So, you will get a string which has unequal number of a(s) b(s) and c(s). You will get a stringwithunequalnumber of a(s) and b(s) and c(s) which is a contradiction. Therefore, L is not aCFL.Thus we can use the pumping lemma for context free languages to show thatcertain languages are not context free.

(Refer Slide Time: 58:37)



We can use it for languages like a powern squared n greater than or equal to oneor a power two power n n greater than or equal to zero,and soon.Thus you have seen the pumping lemma for regular sets, and also the pumping lemma forcontext free languages.

One thing you have to note is weuse the automaton, the finite state automatonfor regular sets.And, we use the contextfree grammars for proving the pumping lemma forcontextfree languages.Both are similarin the sense thatyou can pump a portion, if it is a regular set, you can pump two portions simultaneously for a context free language.