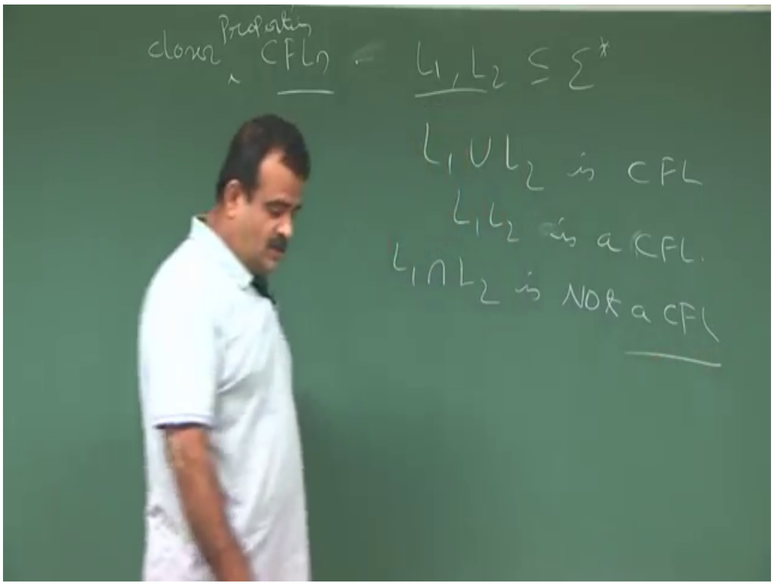
Introduction to Automata, Languages and Computation

Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

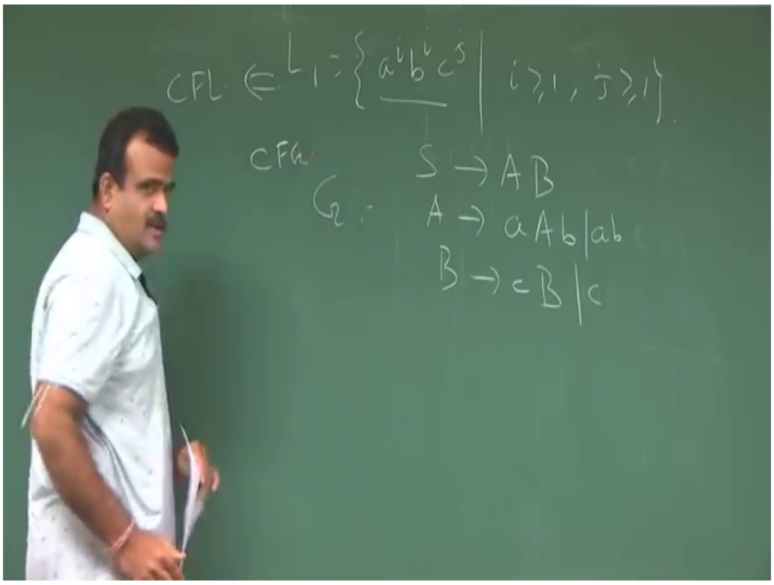
Lecture - 58
Closer Properties of CFLs

(Refer Slide Time: 00:22)



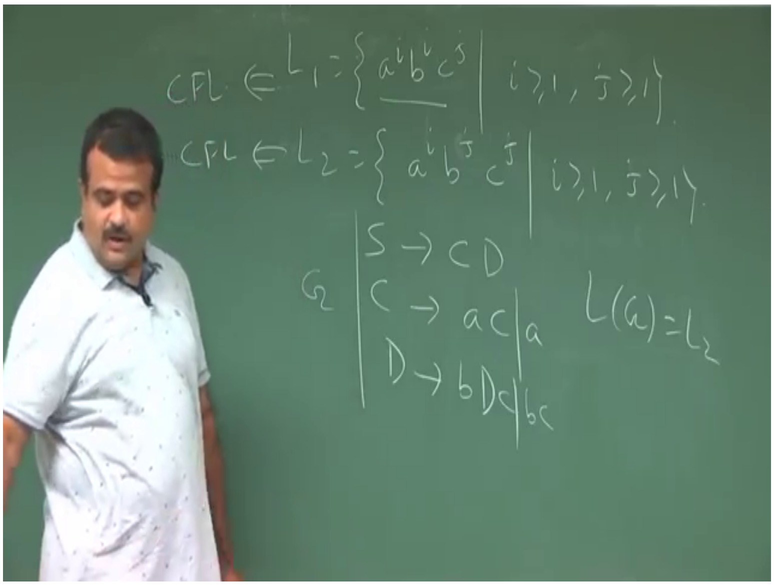
So we are talking about the Closer Properties of CFLs, CFLs are closure properties CFLs. Like if we have given to L 1, L 2 then we have seen in the last class that L 1 intersection union is also a CFL, if this two are CFL. And we have seen L 1, L 2 is a CFL. So and we have seen that intersection; intersection need not be a CFL. So, CFL is not closed under intersection. So, for that we have taken one example, where the intersection is not a CFL.

(Refer Slide Time: 01:24)



So, we have taken L 1 as a to the power i b to the power i and c to the power j where i is greater than equal to 1, j is greater than equal to 1; so this is a CFL. Why, we can have a grammar for this; we can construct a grammar for this like S is going to A B, and then A is going to a A b and or a b; and B is going to this is c B or c small c ok. So, if you follow these rules this is the grammar G, which is having these properties. If you follow this rule, you can see this is accepting all the strains of this. So, this is a CFL, because we can have a context free grammar which can generate this.

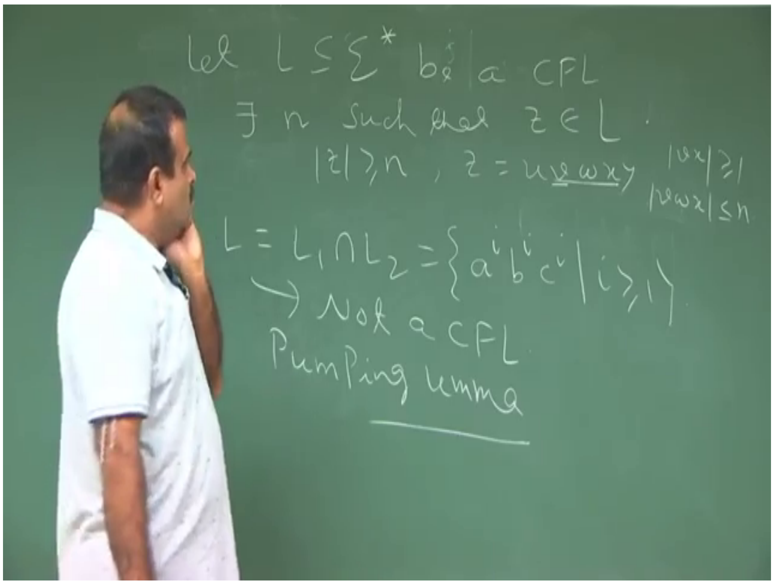
(Refer Slide Time: 02:43)



And also we construct another we take another L 2, which is also a CFL L 2. L 2 we take like this a to the power i b to the power j and c to the power j, where i is greater than equal to 1, j is greater than equal to 1. This is also a CFL we can easily construct a grammar for this.

I have the example. So, if we take S is going to C D capital C and capital C is going to a C a capital C, and D is going to this is b D small c or b c ok. Now, if we have this one then we can easily check it is generating the grammar like this; I mean it is generating the string like this. So, it is basically the generating this grammar. So, if this is our G, then L of G is L 2, so that means, this is a CFL. Now, what is the intersection of this?

(Refer Slide Time: 04:02)

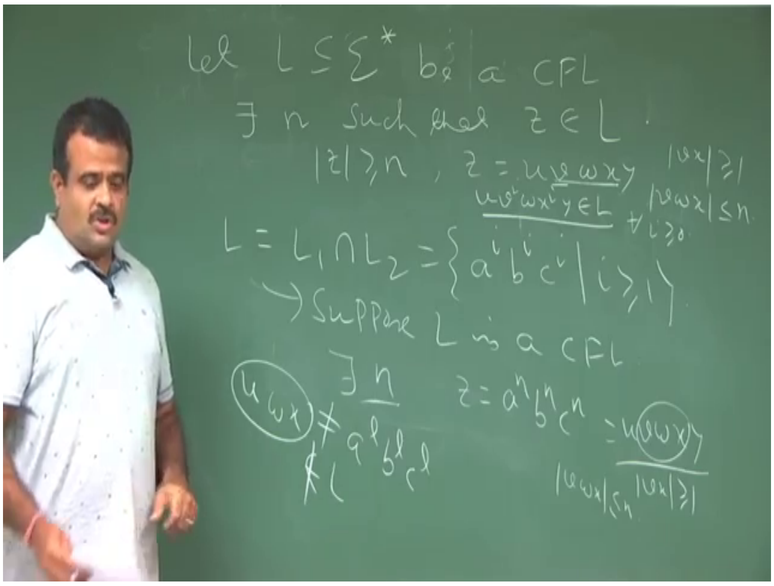


Now, intersection of this is nothing but; L 1 intersection L 2 is nothing but a to the power i b to the power i and c to the power i, where i is greater than equal to 1, where i is greater than equal to 1. So, this we have seen this is basically of the form that yeah, this we have seen this is not a, so this is say L this we have seen not a CFL, why? Because for this we have used what is called pumping lemma for CFL. So, let us just recap the pumping lemma.

So, if we this is a necessary condition to a grammar to a language to be a context free language. So, if we have say let us just recall the pumping lemma for CFL. So, given a language which is a CFL be a CFL, then there exist a n such that if we choose a string from this I mean, if we choose a string from this grammar such that length is greater than n ok then we can write this string as u, v, w, x, y where this and this v and x must be greater than equal to 1 and this one must be v w x must be less than equal to n ok.

So, this is the; and the total is greater than n, the total length these are all substring of the string. Now, so there has to be n, so we assume; so, suppose this is we assume this grammar is this is a CFL.

(Refer Slide Time: 06:15)



Let us assume suppose this will reached over a contradictory. Suppose L is a CFL. Then there has to be a sum n exists for this where if we take a any string of length more than n that should be written as this and this is telling we can pump here, this is telling u v to the power i w x to the power i y this must also belongs to L. And this is true for all i greater than equal to 0 that is the property; this is the necessary property for a grammar to be a CFL.

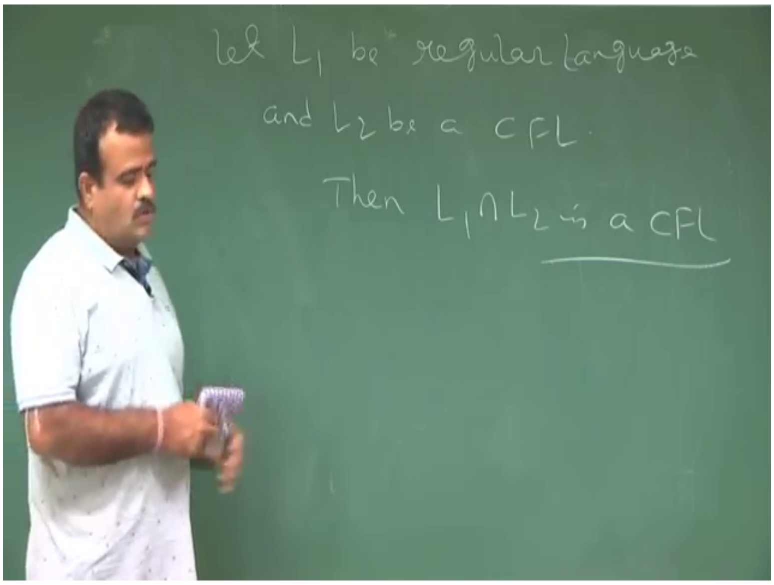
So, suppose this is a CFL that means, if there exist a n, so that n we choose to be I mean any n will work; now we choose a z which is more than n. So, we can n to the power n b to the power n c to the power n, this is also belongs to L. And this length of this is 3 n which is greater than n; so these we want to write in u v w x y ok. So, now this must be v x must not be a empty string and this length must be less than n.

Now, from here what we can say? We can say this v and x both are not belongs to a and c, because if v is belongs to a and x is belongs to c, then the length will be more than n.

So, they are either in a b or b c. So, if they are in a b; then if we just put this i is equal to 0, then u w y this will be not of the form of a to the power l, b to the power l c to the power l. So, then this will not be belongs to l.

So, there is a contradiction even for i is equal to 0, then this is not a this our assumption is wrong. So, this language is not a CFL, so that means, this given a so, intersection is not always a CFL of given to CFL. We will take more using this property we will have more example, but before that let us just; see where this can be a CFL.

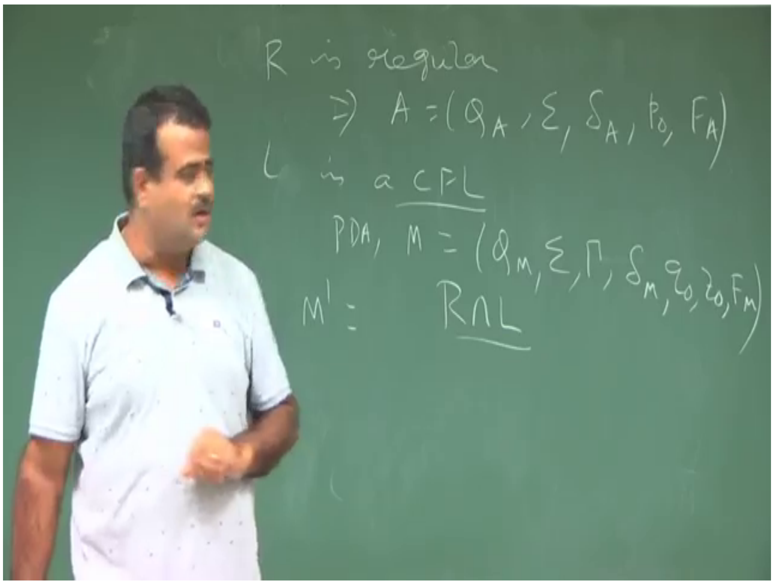
(Refer Slide Time: 09:02)



I mean given suppose let L 1 be a regular language need not be a C FL; a regular language mean it is CFL, but not all CFL is a regular language and L 2 be a CFL. Then we will see if L 1 is also CFL we know intersection is not a regular. But if one is regular, another one is CFL, then intersection is then L 1 intersection L 2 is CFL ok; so this proof we will do now.

So, we state this in the last class, we have not proved that, so this proof will do here. So, intersection of a regular language and a context free language R is a context free language; so let us try to prove that.

(Refer Slide Time: 10:07)

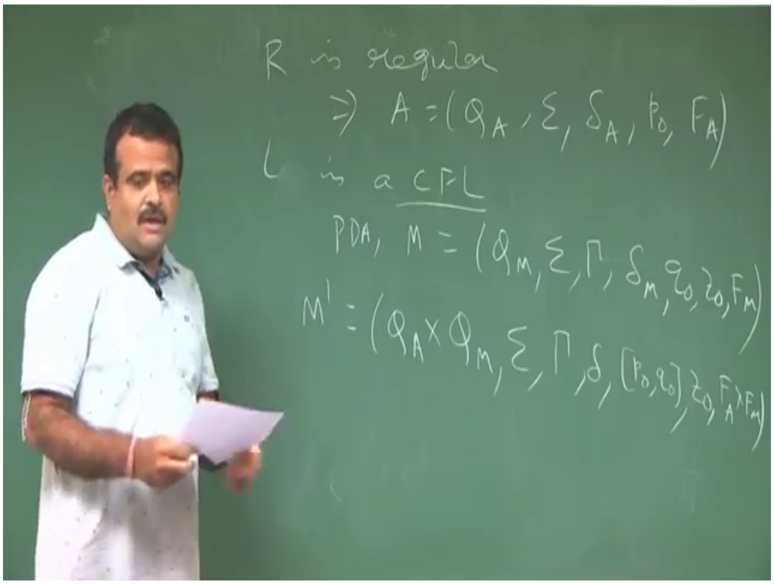


So, L 1 is regular. This implies, we have a DFA for this say A; Q A, sigma, delta A and p 0 is the starting state of this DFA and final state F A ok. And L 2 is a CFL that means, there has to be a context free grammar, which is generating this that means, there should be a PDA Push-Down Automata which is accepting this either by the means of final state or by the means of empty state.

So, we will here take about final state so that means, we should have a PDA, M such that this is Q M, sigma; sigma is the input alphabet, this is delta M, q 0 is the starting state, z 0, F M; F M is the final state of the period. So, we know the; this is accepting by the means of final state ok.

Now, using these two we want to construct a PDA M prime which will accept this. So, this is a R which will accept this is a L, which will accept the R intersection L. So, basically we will take the intersection I mean will run this; we will take the state intersection Cartesian product. So, there for the new PDA that is I am giving you the idea of this.

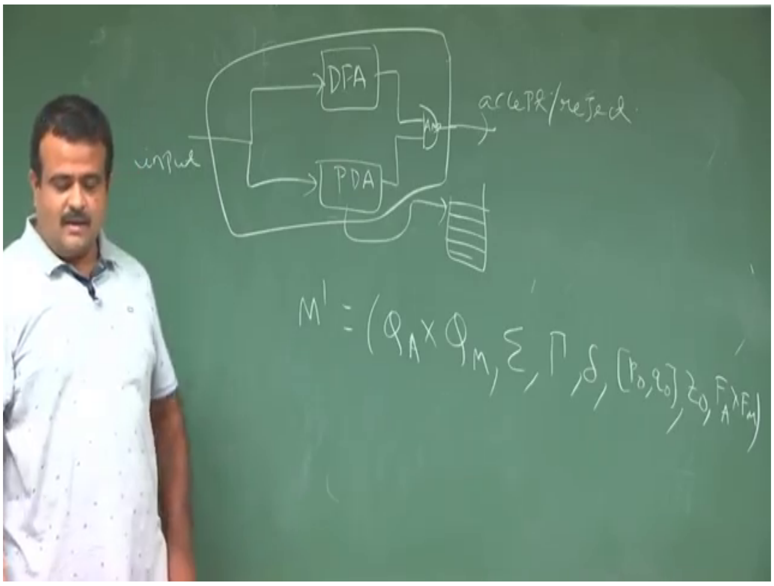
(Refer Slide Time: 12:11)



So, for the new PDA we have this yeah so, let this is our Q A cross Q M so, it is a Cartesian product and this is sigma is same and the stack symbol is same, because stack only exist for the PDA, there is no stack concept for the A.

So, stack symbol is same and you have to different delta transitional rules and the starting state are p 0, q 0. And this is the starting stack symbol and the final state is F A cross F M. Now, the idea is so we are running this two automata in parallel. So, one automata; so the concept is we are running this two automata in parallel.

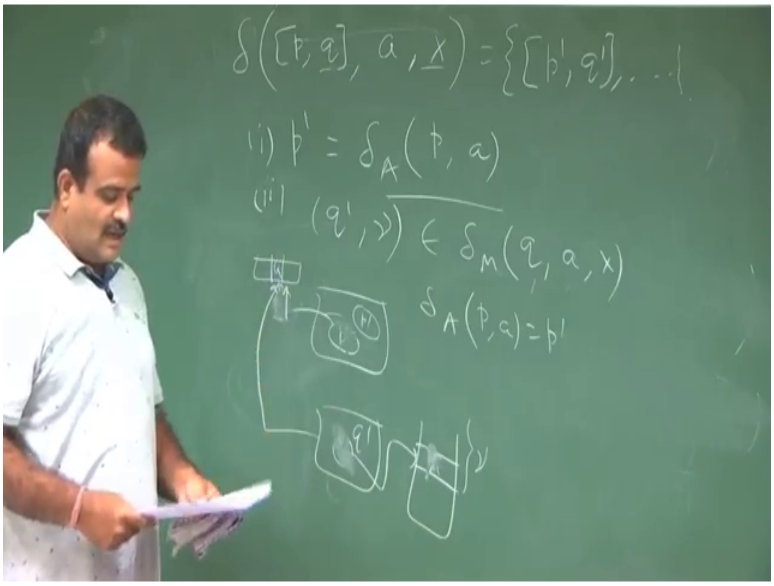
(Refer Slide Time: 13:09)



So, this is our DFA and this is our PDA and this PDA we are having a stack over here and this we have a input we are running this in parallel. If both are accepting then we have a AND gate over here. So, accept or reject this can be categorized by 0 or 1 ok. If give an input string, if it is accepted by the DFA as well as if it is accepted by the PDA, so if the both cases it is 1, then it will it is accepted. If one is accepted another one; if one is reaching to a final state another one is not reaching to a final state, then it will not accept and this is the stack symbol.

So, this is the way this is accept or reject, this is input string. So, one input string will run the automata; I mean finite automata and will run the PDA we for the PDA we take the move for this is stack also. And then we if we reach to the final state for both the cases, then the answer is I mean then we will say yes, yes, then it will pass this AND gate and it will say yes then it is accepted; otherwise if one is rejected, one is not accepted, then it would not pass I mean, if one is 0, then this is 0. So, this will be rejected; this is not accepted. So, formally the PDA is like this so the delta is so you have to define delta for this.

(Refer Slide Time: 15:05)



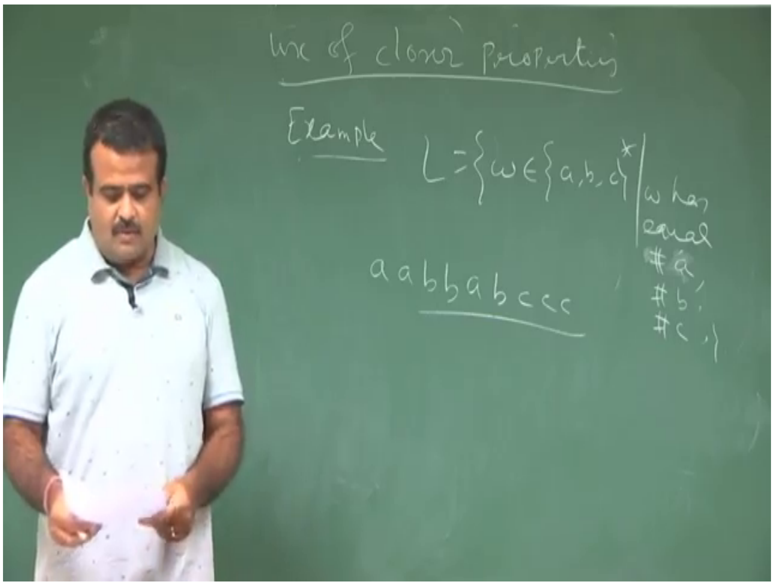
So, this delta we are going to define. So, delta is input is so we are taking p and q; two state one state is coming for the DFA, another state is for the pushdown automata. And we have a input of tape and we have x is the state symbol. So, this is defined to be a this is the set of all pairs like this, p prime, q prime dot dot dot where so p prime is nothing but delta A of p comma a. So, this is a deterministic finite automata so, you have only one state to go; so p prime of this.

And this q prime, so q prime gamma; gamma is the I mean state see this must belongs to delta m of q, a, x. So x is the this symbol so that means what? That means we are say in the DFA; this is the DFA control so, we are at p and in the tape we are reading a S A. So, this is reading a and in the PDA what we are at? We are at q. So, we are at q and our top of the stack is x and we are also reading this a. So, this is running in parallel.

Now, with this move if suppose this is going to p prime suppose delta of A p comma a is p prime. So that means, this will go to next and this will go to p prime this control and here also it will go to q prime and this will be replaced by gamma that is the idea ok. These two systems are running in parallel. And if one I mean if both the system are reached to the final state, then only we say this is reaching to a final state. So, this is the rough idea of the construction of this ok.

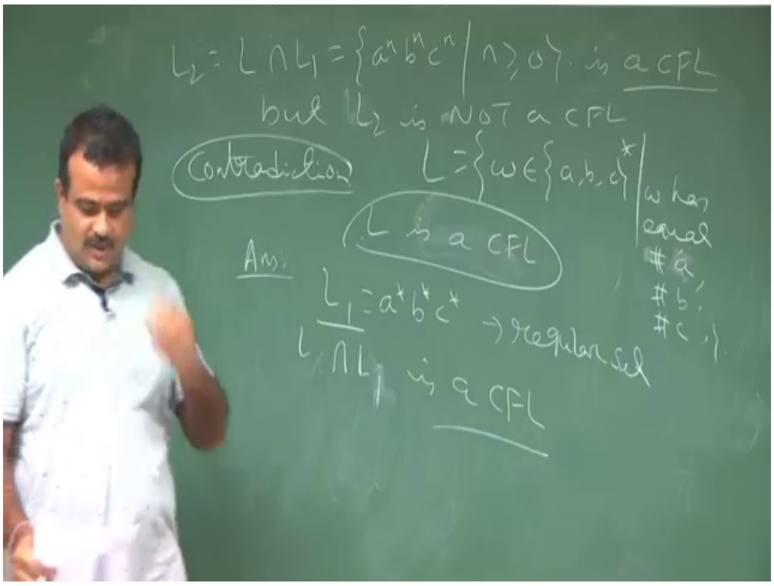
So, now we will talk about some applications of the pumping lemma and the closure property mixer. Use of closure property, we can say certain using a pumping lemma also certain language are not regular ok.

(Refer Slide Time: 17:52)



Ok, say for example, so we want to use of closer properties and pumping lemma. Both we combined and we will see some example, where we can show certain language, they are not CFL. So, we will use the pumping lemma for the CFL; so let us take an example. So, say w, so it is a coming from a, b, c such that where w has equal number of a's, equal numbers of b's and equal numbers of c's ok. I mean it is a, b, c. So, like if we have say a a b b a b, then c c c this is belongs to because how many a's are there? 3 a's, 3 b's, 3 c's equal number of a b c's ok.

(Refer Slide Time: 19:12)

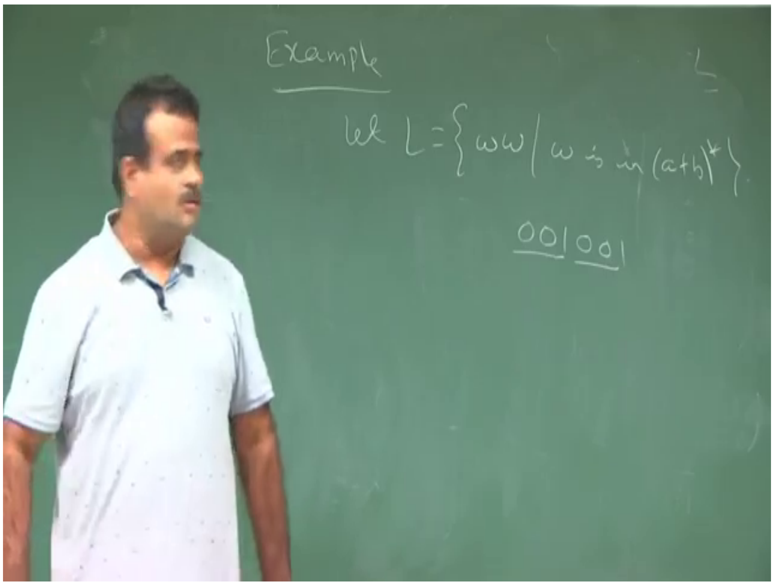


So, now we will see whether this is a CFL or not. So, the question is whether this is a CFL or not? Ok, so to show this we have used the will see this is not a CFL. So, to prove this we will use the closer property and as well as the pumping lemma. So, let us try that ok. So, how to pass it solution? So we take L 1 a star b star c star. Now, this is a regular expression if you recall the regular expression so, this is a regular language, this is regular set. That means, any number of a's followed by any numbers of b's followed by any number of c's; so this is a regular set ok.

Now, if we take this intersection with this, suppose this is a CFL so, we are assuming L is CFL. Now, if L is CFL then L 1 intersection L 2. L 1 is regular this will be also CFL just now you have seen the construction. Now what is sorry what is L intersection L 1, now what is L intersection L 1? That we have to check; so that is nothing but this set. So, L intersection L 1 is a to the power n b to the power n c to the power n that we can easily verify ok. So, equal number of ones, then this is any number of equal number of a's equal number of so, this is any number of a's, any number of b's, so this will be like this. So, this is basically for n greater than equal to 0 this is the intersection.

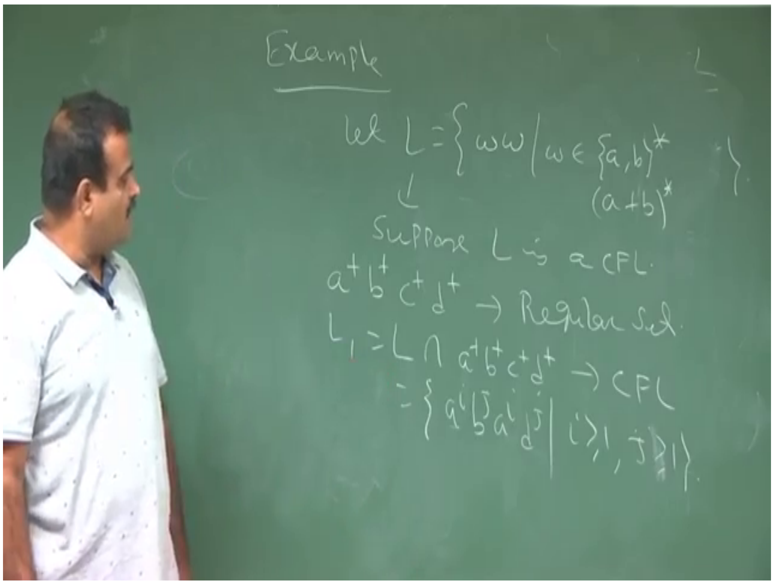
So, now if we assume this is CFL, then this will be CFL? But this is not a CFL we have seen using the pumping lemma, but this is say L 2; L 2 is not a CFL. Hence contradictions;, we reached to a contradiction; so we cannot assume this is a CFL so, this is not a CFL. So, that is using the contradiction we can say this is not a CFL ok. So, yeah now we can take other example, using the properties and there are other closerness of this using the; what is called the yeah.

(Refer Slide Time: 22:10)



So, we can take another example. We can use the substitution method to see whether a CFL is closed under substitution, we will have a mapping I mean 0 is going to a say, 1 is going to b and like this. So, we will have a mapping and we will see that the CFL is close under substitution. So, let us take another example. This we have seen w w w is in a plus b star ok. So, this is basically pair like this, I mean w is any string so, 0 0 1 then 0 0 1 like this; this is w, this is w like this. So we will show that this is also not a CFL, so how we can show this?

(Refer Slide Time: 23:13)



So, this is basically w is coming from w is any string from a b; this we can write and regular expression a plus b star this we know this is the regular expression ok. So, how to show this is not a CFL? So, for that we will use the; that again the same property like a intersection of a regular set on a CFL is a CFL. So, we will check this.

Suppose, this is a CFL ok now, we take this a plus b plus then c plus d plus this is the regular set. Now, we take L 1 which is L intersection; this L intersection a plus b plus c plus d plus. Now, this will be a CFL, because this is a regular set and we are intersecting with the CFL. So, this intersection will be CFL on regular set intersection with the CFL is give us a CFL.

Now, what is this set? This set is nothing but a to the power i b to the power j a to the power i and d to the power j, i is greater than 1 j is greater than 1. Now, we know this is not a CFL, so that means, I mean we can use this by a substitution method and then pumping lemma to show this is not a CFL. Then we can claim that this is reach to a contradiction that this has in from is a CFL so, this is not a CFL. So, this is the way we can prove that some of the language we can take the example some of the languages not a CFL.

Thank you very much.