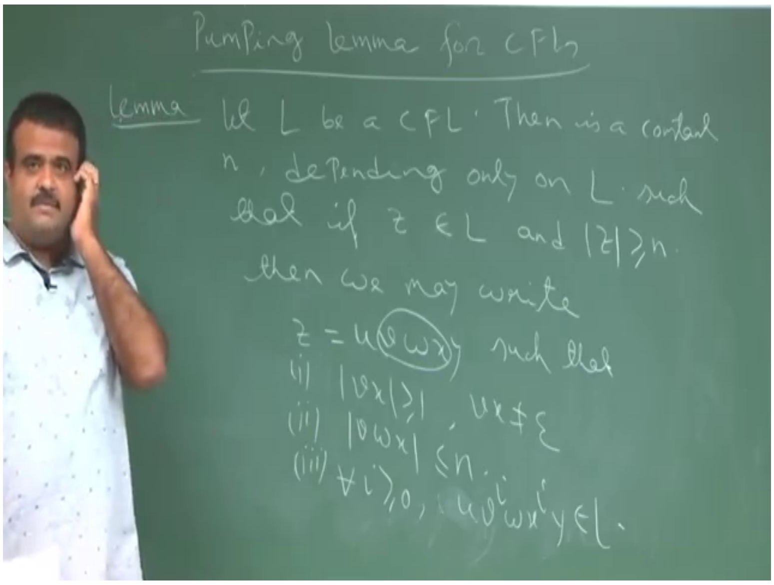
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 57
Pumping Lemma for CFLs

So, we talk about the Pumping Lemma for Context Free Language. So, it is a we know the pumping lemma for regular language. So, that is a necessary condition to be a language to be a regular, even using that we have seen many languages non-regular. So, that is a good way to prove that in language is not regular. If any language is not regular, the pumping lemma will help us to prove that because this is a necessary condition.

(Refer Slide Time: 00:53)



So, similarly we have a version of the pumping lemma for CFL; so let us define that. So, this is the statement let L be a CFL; Context Free Language, then there exist a n there is a constant n depending only on L such that if we take a z in L whose length is more than n; if you take a z in L mean L and length of z is greater than n ok. This is very similar to the pumping lemma for the regular language.

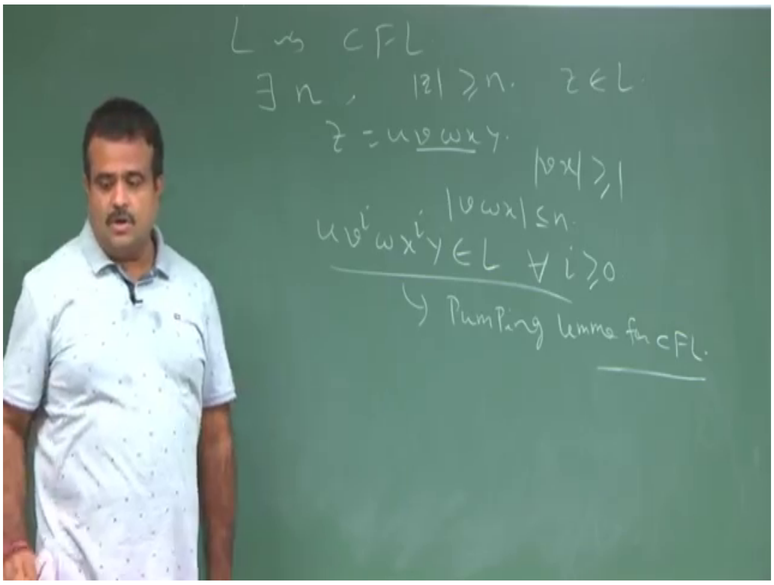
Then, we may able to write we may write z as uvwxy. These are all strings coming from that input alphabet such that we are going to pump here and here such that these two should not be I mean epsilon; that means, I will write that such that three condition is satisfied such that this vx should be greater than equal to 1. So, that means, they are not in all string vx is not equal to epsilon they are not a null string, because we are going to pump there.

And, second condition is vwx is less than equal to n ok. This part is less than equal to n, then we have this like then this is the condition necessary conditions that for all i greater than equal to 0, you can pump there I mean u v to the power i; u v to the power i w x to the power i y, this belongs to n. This is for all i including i is equal to 0. If i is equal to 0, this will not be there; uwy will be there, but if i is 1, that is our z; if i is 2 that is this is the very similar to the pumping lemma of the regular language what we have; we can pump there ok.

So, this is the statement the proof will be there in the lecture note. I am not going to discuss the proof in the class. Proof is little tedious like this n is basically if we draw the tree with the number of way this is CFL. So, you have a number of variables. So, 2 to the power of n, so, if you draw the tree of parse tree; so that is the how much it is maximum size is 2 to the power of L. So, that is nothing, but our this n ok. So, proof will not be there in your syllabus for the exam.

So, this is the statement. So, now, using these we can; so many language which are not regular.

(Refer Slide Time: 04:55)

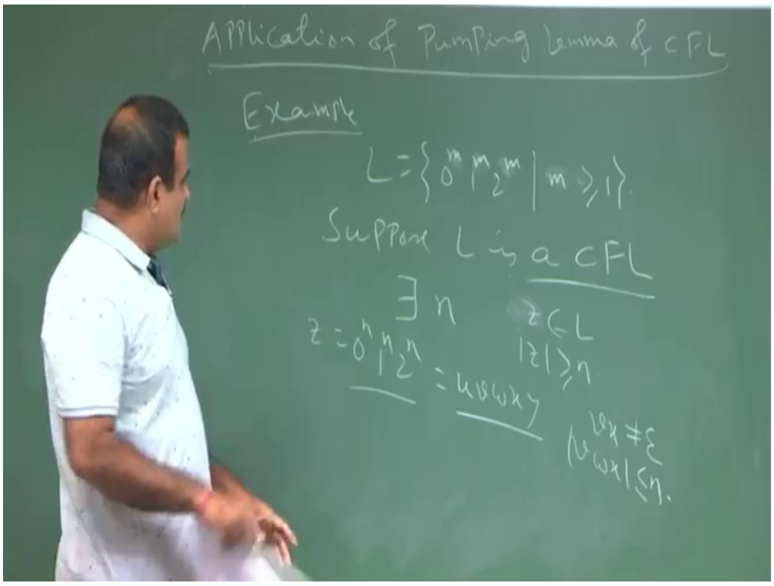


So, in short we can write this. So, if L is CFL, then there exist a n such that if we if z is greater than 0, z belongs to L and then z will be written as u v w x y such that this is greater than 1 because this should not be epsilon and this is less than n v w x is less than equal to n and total is greater than n.

And, then we can pump there then this is telling us there is the necessary condition then u v to the power i, wx to the power i. We are pumping in x string by v string and x string y, this will be belongs to L; for all i greater than equal to 0 for all i greater than equal to 0. So, this is what is called pumping lemma for CFL ok.

So, now we will use this to see some language which are non CFL I mean. So, let us take one example where we take a language and we prove that it is non-CFL; so further we will use this pumping lemma.

(Refer Slide Time: 06:35)

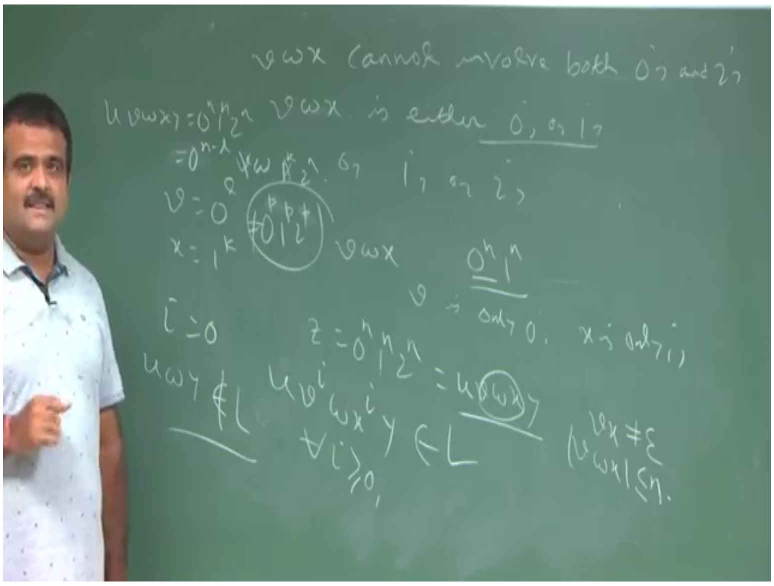


So, this is the applications pumping lemma of CFL ok; so how to; so we will check whether certain language are regular or not. For example, if you take a language like this 0 to the power n, 1 to the power n, 2 to the power n or a to the power n, b to the power n, c to the power n anything; where n is greater than 1. Is this a CFL? No, this is not a CFL; this is not a CFL. We have to, but we have to prove that; so to prove that we will use pumping lemma.

So, what we do? Suppose, this is a CFL; let us assume this is a CFL. Suppose L is a CFL, this is the assumption and we will reach to a contradiction. So, if this is the CFL means there exist some n ok, we take this m say otherwise this m and this n will be confusing. So, there exists a n which is in the pumping lemma that such that if you take a z in L and if you take mod of z greater than equal to n. So, if you take just z equal to 0 to the power n, 1 to the power n, 2 to the power n. So, what is the length of this? 3 n; 3 n is greater than n; so this is our Z.

Now, this z will be written as uvwxy, where we have all the properties like vx is not equal to epsilon. So, that means, this is greater than length of this is greater than 1 and vwx length is less than n ok. Now, since this length is less than n yeah; so we have to argue here. What do you mean the argument over here? So, we are taking this.

(Refer Slide Time: 09:13)



Now, the claim is vwx. vwx, this part this cannot involve both the 0's and 2s. This cannot involve yeah this cannot involve both 0's and 2's ok. Why? Because this is less than n, if it is has to involve. So, this is basically 0 0 0 n times, then 1 1 1 1 n times, then 2 2 2 n times. Now, if you have to involve 0 and 2, then this will automatically come in the middle. So, there this will create the length more than n, but we are taking this uvx is less than n; so, it is either involving 0 1 or involving 1 2.

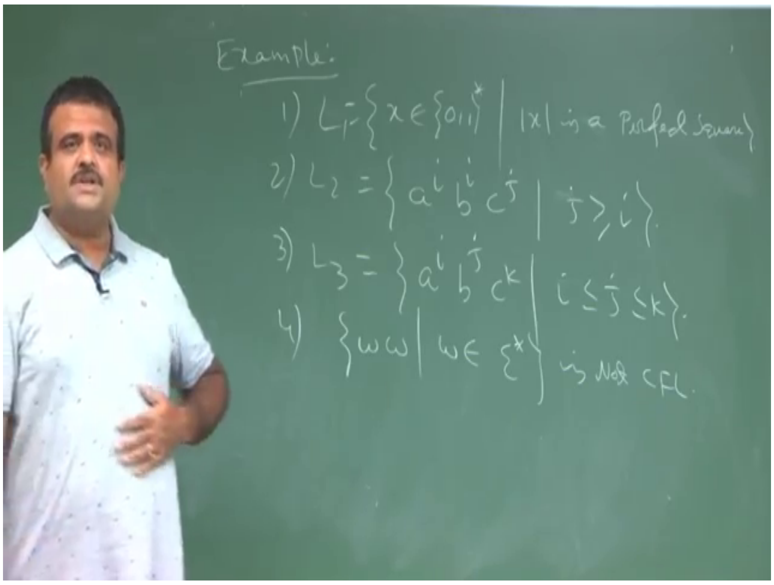
So, that means, uvw is either 0's or 1's or 1's or 2's ok. So, if it is 0; 0's or 1's so that means, uvw is coming from here so; that means and they are less than n. So, the v and so, that means v and so, this uvw sorry vwx they are basically coming from this 0 to the power n, 1 to the power n or 1 to the power n 0 to the power n. So, if they are coming from this and their length is less than n so, we have few part over here, few part over here; that means, the v v will consist of 0's; v is only 0's and x is only 1's ok. So, that means, v is some 0 to the power L and x is some 1 to the power k something like that ok.

So, that means, what is u? u will be so, what is uvwxy? uvwxy is 0 to the power n, 1 to the power n, 2 to the power n. So, this is basically this will be written as so, u so, 0 to the power so, even though we can have mixture also. So, they are; that means, it is basically 0 to the power u 0 to the power n minus L, then v, then w and then x; x is 1 to the power k and then 2 to the power n something like that ok.

So, now we know that pumping lemma is basically what it is telling? Now, if this is this, then u v to the power i wx to the power i y. This will also belongs to L, this is the necessary condition; for all i greater than equal to 0. Now, if we put i is equal to 0 then it will be like this. So, this part is gone this part is gone. So, then it will not be of form 0 to the power some p 1 to the power p, 2 to the power p, it will not be this form. So, the; that means, even for i is equal to 0 this will give us uwx this does not belongs to L because this is not of this form sorry z this.

So, there is a contradiction; contradiction means this has to be belongs to L for all n I all I, we are pumping there. So, this we reach to a contradiction; that means, our assumption was wrong; that means, this cannot be a regular language, ok; so this is one example, we may have another example.

(Refer Slide Time: 13:50)

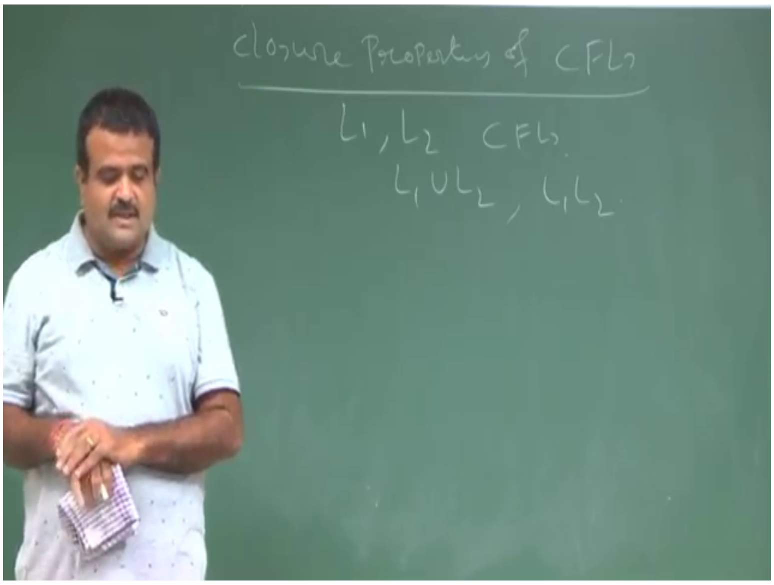


So, I will take another example, but this proof is there in the lecture note for other example we will just state that, more examples. We can prove using pumping lemma similarly if we take a L like this x belongs to such that mod x is a perfect square. This is not a regular; so this is proof will be there in the lecture note, I am not.

So, even a to the power i, b to the power i, and c to the power j; a, b, c are alphabets where j is greater than i this is also not regular; so all these we can prove using pumping lemma; a to the power i, b to the power j, and c to the power k such that i is less than j less than k. So, these all we can show this not a regular language even this w; ww. So, this is say ww same string. I mean w is coming from is not regular is not CFL. So, this all we can; so they are not CFL there many others example are there this will be given in the lecture note, even using the 4 if. So, these are all example will be there in a lecture node.

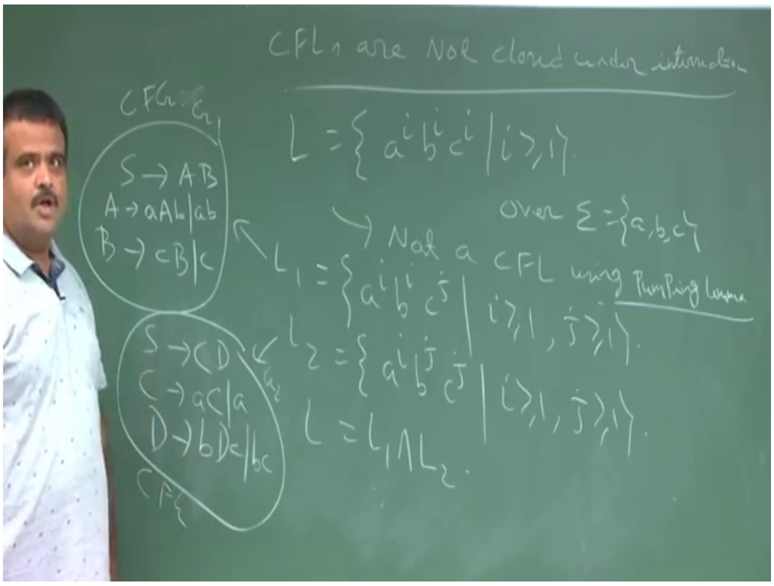
Now, we are going to use this property to show that given two regular language; their intersection is may not be regular.

(Refer Slide Time: 16:03)



So, we have discussed this closure properties of sorry CFL; closure properties of CFL. So, we have seen if we have two L 1, L 2 CFLs, then we have seen that their union is CFL, their intersection their concatenation a CFL. Now, we will see whether their intersection is also CFL is not. So, intersection is not a CFL then CFL. So, CFL is not closed under intersection; so that we will see.

(Refer Slide Time: 16:48)



So, are not closed under intersections ok. So, for that we take a language L; a to the power i, b to the power i, c to the power i where greater than 1. So, this language over a over a sigma a, b, c. Now, we can show this language is not regular using pumping lemma sorry is not a CFL using this we can easily verify. So, these properties this we will use; so this; the intersection is not a CFL.

Now, so, this sorry this is we have to see these language is the intersection of two CFL. So, how to see that? So, we take two language L 1 which is a to the power i, b to the power i and c to the power j; where i is greater than 1 and j is greater than 1. And, if you take another language L 2 which is a to the power i, b to the power j and c to the power j where both the i and j are greater than 1.

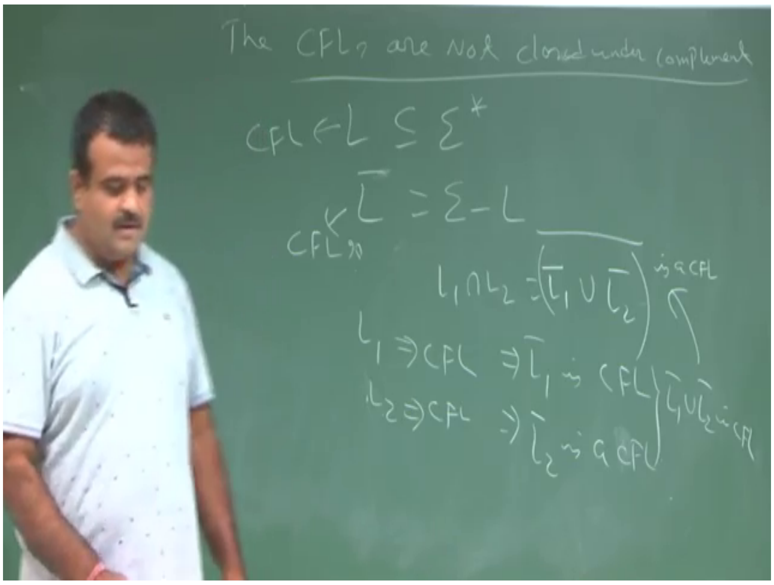
Now, we can just check this L is nothing, but L 1 intersection L 2 then we have this. Now, the question is; so this is L is intersection now these two are CFL how to prove that? So, to show that we need to create a grammar if a CFL means we have a grammar which can generate that, we have a context free grammar which can generate that; so let us construct a grammar for this L 1.

So, you can construct a grammar for L 1 like this. S is going to AB and A is going to aAb or ab and B is going to cB or c. So, this is the grammar for L 1 and what is the grammar for L 2? S is going to CD C capital C is going to aC capital C a and D is going to bD small c and bc small c. So, this is the grammar corresponding to L 2 ok.

So, then we can just; so this is this two are CFL because we have a CFG one CFG which is CFG which is G 1 and we have another CFG context free grammar which is G 2 which is G 1 is generating this, G 2 is generating this. So, that is why this is; these two are context free language and once these two context free language, then we can their intersection is this one, but this is not a context free language. So, if that is why if this is not a context free language we can see using the pumping lemma.

So, the intersection this is the counter example where we have seen the intersection is may not be a context free language always ok. So, this is one example.

(Refer Slide Time: 21:22)



Now, we will see the complement. We see the complement whether complement can be a context free language or not.

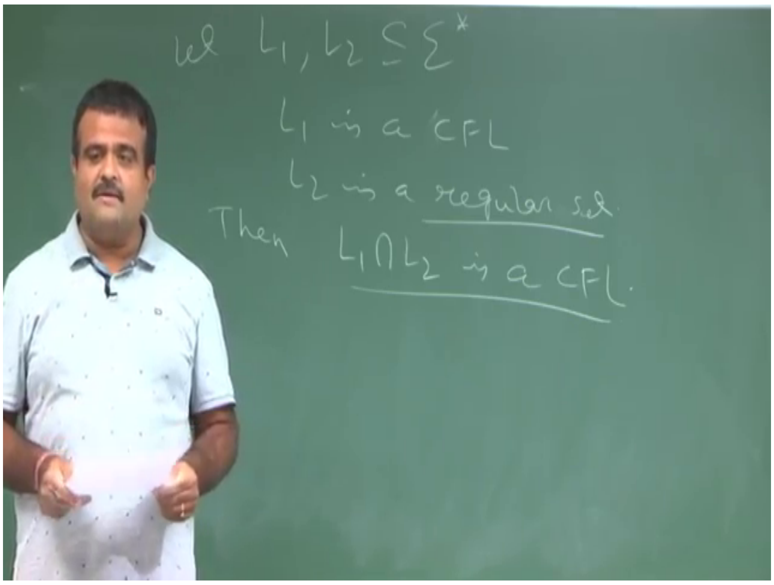
So, the CFL are not closed under complement are not closed under complement; that means, given a language whether the complement is also a CFL or not. So, that means, if L is a CFL which is coming from this here then L complement or L bar is this a CFL, this is CFL.

Now, the question is this a CFL? Answer is again, no and again we can justify this using this intersection property. So, complement for complement what we can do? Suppose, L 1 and L 2 we have. Now, L 1 and L 2 intersection we can see as a L 1 complement union L 2 complement, then whole complement. Now, if we assume that complement is complement is closed under if we assume is the CFL is closed under complement.

So, if L 1 is a CFL and L 2 is a CFL; now, if we assume this L 1, L 2 are complement of CFL that will give us that L 1 bar is a CFL, this is a assumption suppose and we will reach to a contradiction and L 2 bar is a CFL. Now, once they are two CFL, then we know their union is CFL. So, L 1 bar union L 2 bar is a CFL. Now, once these two are CFL once this is a CFL; now we have assumed the complement is CFL, so that will give us this is a CFL, this complement. Now, this is nothing, but the intersection.

But, we know the intersection is not a CFL. I mean if we have two language; L 1, L 2 which are CFL their intersection need not be a CFL; so that is the contradiction. So, the complement is not always a CFL because this property, but now we will see another interesting property of the intersection although the intersection of two CFL is not CFL.

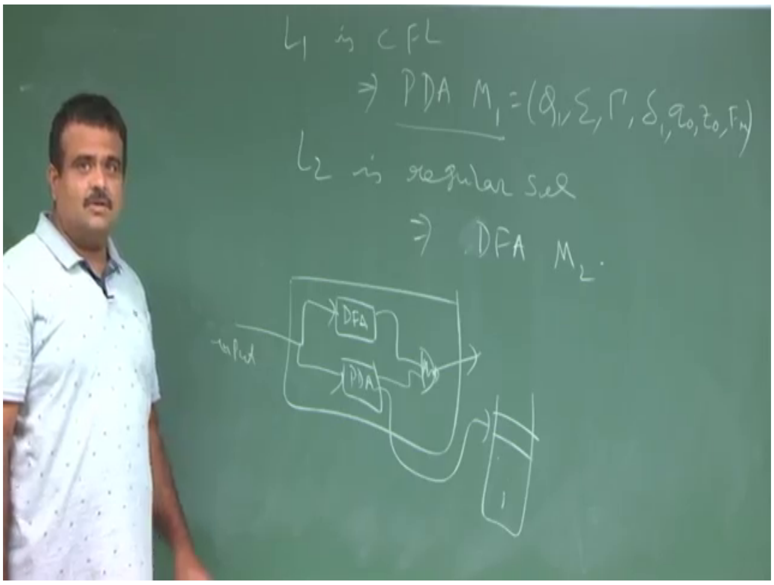
(Refer Slide Time: 24:34)



But, if we take one CFL and another one is regular, then their intersection is CFL. Like if we have two language L 1 and L 2 and suppose L 1 is a CFL and L 2 is a regular is a regular set, then that means, then we can show that L 1 intersection L 2 if it is CFL we know this is not a CFL if L 2 is CFL, then L 1 intersection L 2 is not a CFL.

But, if it is regular set, then we can prove that this is a CFL. The CFL is the so, CFL is not closed under the intersection, but once we take one language is CFL, another language is regular then their intersection will be CFL. So, this proof we are going to just give the outline of the proof.

(Refer Slide Time: 25:56)



So, basically what we know? If we have a if L 1 is CFL; that means, it will corresponding to some PDA M 1 which is Q, sigma, Q, sigma delta of delta 1, q 0, z 0, F m. Anyway we have a M 1 which is PDA and for L 2 we have a L 2 is a regular set. Then we have corresponding a DFA, Deterministic Finite Automata, M 2 which will accept that.

Now, what we do? We run these two in parallel. So, we will construct a new PDA where we do not have stack for this, but we have stack for this. So, we have a this is our DFA and this is our PDA and PDA is accepting this stack. This is the top of the stack like this and this is the input is going here, going there and this we are XORing and we are taking the acceptant or reject; this is the sorry, AND gate not XOR gate.

So, this is the input we are running this in parallel. So, and the stack is with the PDA only. So, this PDA will be running and then once this is. So, this formal proof of this formal construction, we will discuss in the next class.

Thank you.