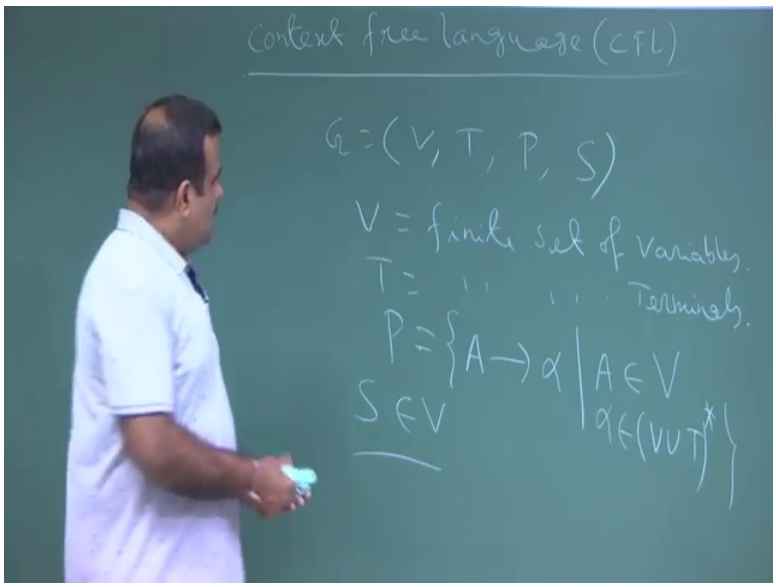
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 37
Context Free Language (CFL)

(Refer Slide Time: 00:17)

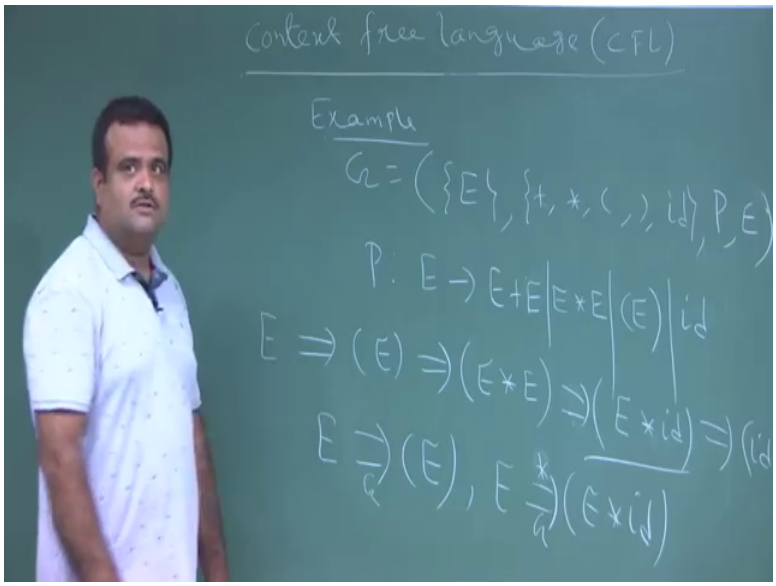


So, we are talking about Context Free grammar so, just to recap. So, it consists of 4 tuple V, T, P and S. And V is the finite set of variable variables and T is also finite set of terminals or we can say alphabets in our automata terms itsa set of alphabets which is finite.

And P is the production rules productions. So, P is will be of this form like this set of all such rules A going to alpha, where A is a variable and alpha is a string who is consist of the variable and the terminals ok. So, all such production rules for all such productions is the P that is the productions. And, this is S; S is the special variable which is a I mean from V which is a start variable we can say ok.

So, this any such 4 tuple is called cortex by gamma. Now, in the last class you have defined the derive say if. So, let us take an example suppose, we have a gamma like this.

(Refer Slide Time: 02:07)



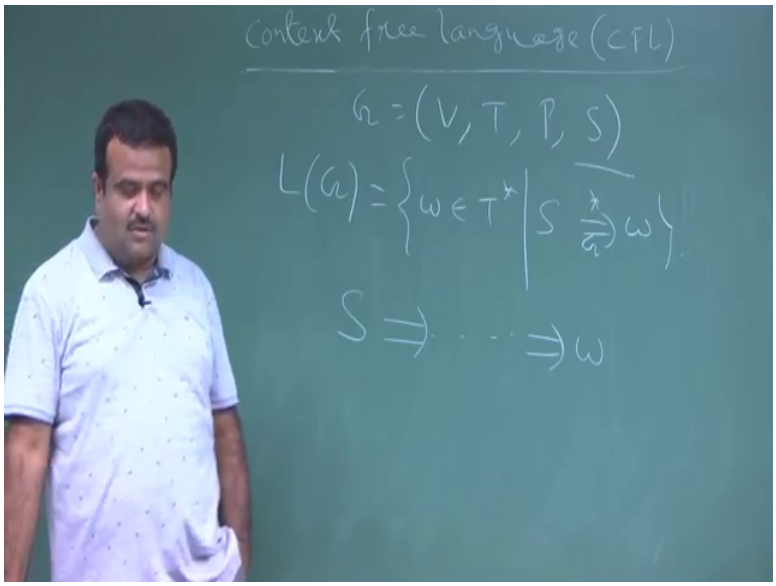
So, we have only one variable and say we have few terminals plus into bracket this bracket id. So, this we have seen in the last class. So, what is E? E is the only variable. So, our S is also E now what is P? P consists of this E is going to E plus E or so, this way you can write we have seen E bracket or id these are the rules ok.

Now, we can take some derivatives like E, E is going to we can take any one of this say E is going to bracket E, this derivative then we can take this E is going to E plus E or E start E, star means into we can say. Then, we can say this E is going to id, then we can sorry E not equal this is the derivative derive. Now, we can say this E again is going to id star id ok.

So, this is the so, if we take this one. So, this is derived from this, but not with the one step, but this E is derived from this is derived from this by one step. So, we can write P is derived this is derived just one step G ok.

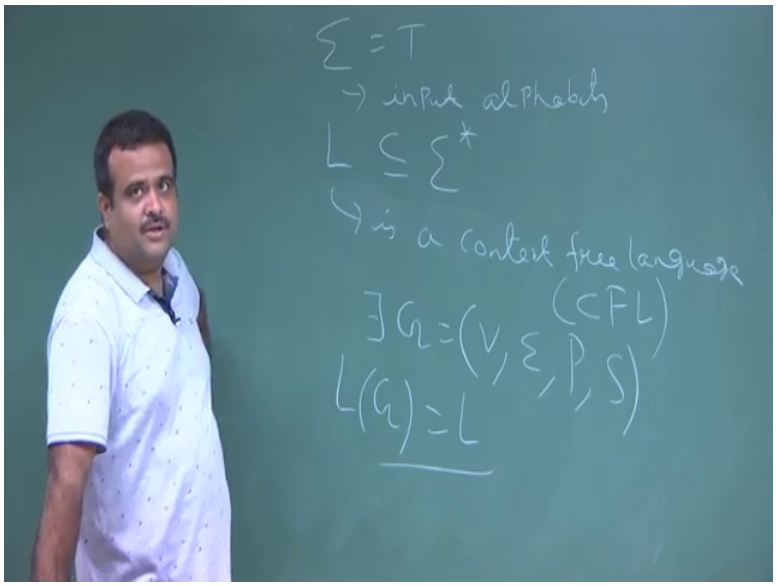
Now, this is derive in 3 step 1 2 3. So, we can say 3 we can write here, this is 4 step or in general we can write star G, G means the gamma rules. So, by this rule we are deriving this. This star means more than one step we can derive this from the starting variable. So, that is the meaning of star. Now, based of the star we can define the language of this grammar or we can say the language generated by G.

(Refer Slide Time: 05:03)



Suppose, we have a G so, what do you mean by language generated by G? So, this is denoted by L G; L G is the set of all strings of terminals, this is coming from T star. Such that from starting variable we should able to reach to by applying this rules from P, we should able to reach to w and w consist of only the terminals w is having no variables so; that means, under this G we should able to reach w. So, all such strings if we collect those strings are basically the collection of those strings is a language, because if we consider these as a alphabet sets its a language. And then this language is called the language generated by this grammar G ok. Now, when we say a language is a context free language.

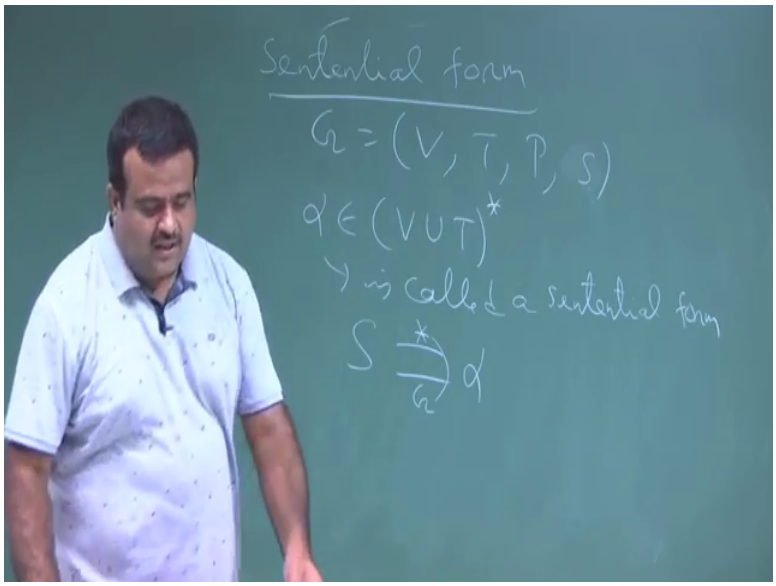
(Refer Slide Time: 06:19)



Suppose, we have given a sigma, which is basically T which is a input alphabet finite set of input alphabet. Now, we take a language we take some collection of the strings coming from this. Now, when we say this is a context free language this one is called is called L is a Context Free Language or in scot it is called CFL. If, we can if we have a context free grammar, which is generating this language, if there exists a context free grammar with the terminal as this with some rules. And the starting stage such that the L of G is the L.

Then, we say this is a context free language ok, this is the definition of a context free language. We should have at least one context free grammar. So, for a given language we can have more than one context free grammar, we will see that by a example.

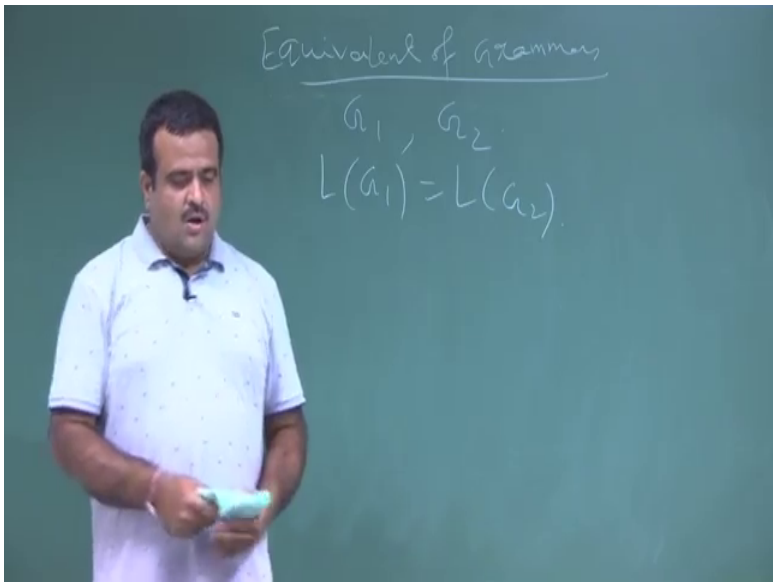
(Refer Slide Time: 07:55)



Now, we will define what is called sentential form; sentential form. So, we have given a context free grammar. So, this is we have given a context free grammar, now alpha which is a string consists of some variables and some terminals ok. Now, alpha is called a same alpha is called in sentential form this alpha is called a sentential form if alpha can be derived from S by the rule of this.

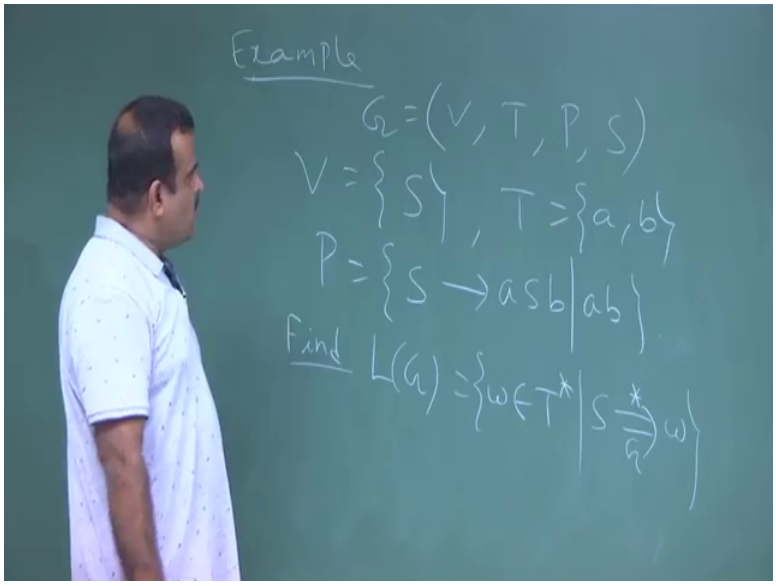
So, if with one or more stiff alpha can be derived homiest, then it is called alpha need not be a terminal alpha may consist of variables also. But, only thing the string alpha should be derived from the starting variable S with the rules of with the productions of this G, using the rules of P we should able to use this, then it is called a sentential form this alpha is called as sentential form ok.

(Refer Slide Time: 09:35)



Now, we can define the equivalence of 2 grammar, when we say 2 grammar are equivalence of equivalent of grammar ok. We have given 2 grammar G 1 and G 2. So, we say this 2 grammar are equivalent, if they are generating the same language, if the language of then we say these 2 grammars are equivalent. So, you will take some example of the equivalence grammar ok. Now, let us take some example and we will see some language of that.

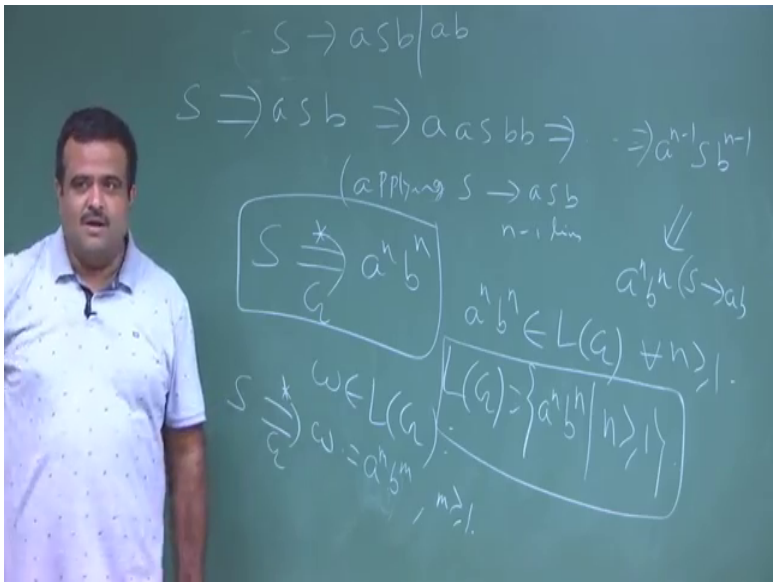
(Refer Slide Time: 10:21)



So, let us take a grammar like this consider a context free grammar V, T, P, S; where P is already having a variable S and T is the terminals for the alphabets which is a b and P the rules. So, we have this rules S is going to a S b or S is going to the two rules ab ok.

We can combine these two and we can write S is either this or ab this is a grammar, this is a context free grammar. Now, we want to see we have to find the language generating by this grammar find L of G, we have to find the language which is generating by this grammar. So, what is the definition of L of G? L of G is nothing, but set of all strings of terminals or alphabet such that, which is derivable from the starting variable S, such that S is S should be reachable to I mean this under G with the derivation of by rules productions P ok. So, we have to find this set L of G. So, what is L of G, now you will just see the S.

(Refer Slide Time: 12:19)



So, our S is of the form a S b or a b ok. Now, form S what we can do. So, form S we can derive you can either apply a b or we can apply this. So, if you apply a S b then again if we apply this S a S b. So, these are sentential form. So, a a S b b. So, if you keep on apply this, how many times n minus 1 times this. So, basically we are applying this production how many times n minus 1 times, then where we will reach we will reach to a this a to the power n minus 1, s b to the power n minus 1 ok. Now, then after reaching this we just apply S is going to a b.

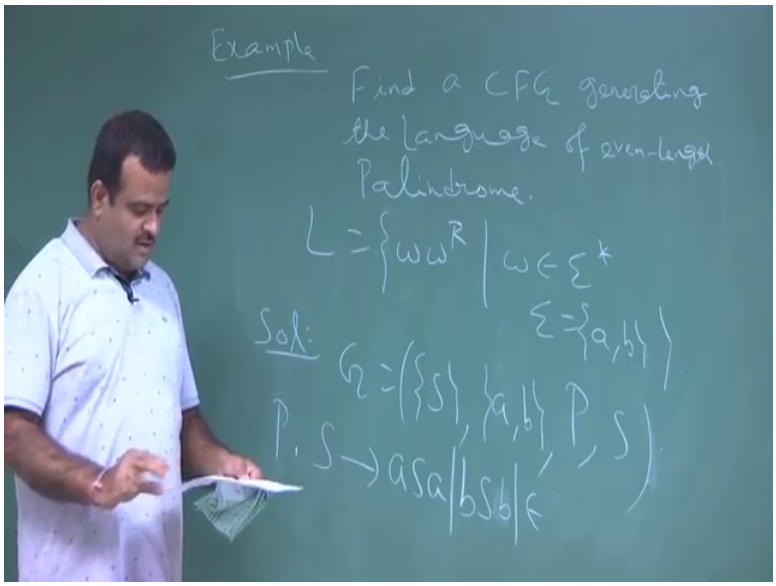
So, if you do this it will give us a to the power n b to the power n. So, a to the power n b to the power n is the sense sentential form so; that means, a S a to the power a to the power n b to the power n is derived from S, for any n if n is 1; that means, we do not apply this rule we will they go for this rule. So, S is apply this is applying 0 times. So; that means, S is from so, this is the ok.

So, so; that means, a to the power n and b to the power n must belongs to L of G for all n greater than equal to 1 ok. So, L of G a to all the possibility to the power n b to the power n will be L of G. Now, the other way around we have to so, that L of G is to so, this L of G is nothing, but a to the power n b to the power n for all n greater than these. So, this is a subset of this we have shown.

Now, we have to prove this is a subset of this; that means, if we have to take any w from this which is the a terminal string of terminal, which is derived from S, then we have to argue that w must be of this form. Why, because our rules is like this. So, either we can apply this S is going to a S b or a b. So, in any case it will be of this will be is the step like this.

So, in any case w will be in this form a to the power m sorry w will be either of this form a to the power m b to the power m. So, this is the, so, we can argue so; that means, this is a subset of this. So, eventually L of G is this. So, this is accepting the language a to the power n b to the power n, I mean this is generating the language a to the power n b to the power n, but we know this is not a regular language. There is no finite automata which can accept this language, but we do have a context free grammar, which is generating this language ok. So, we will take some more example; we will take few more examples.

(Refer Slide Time: 16:27)

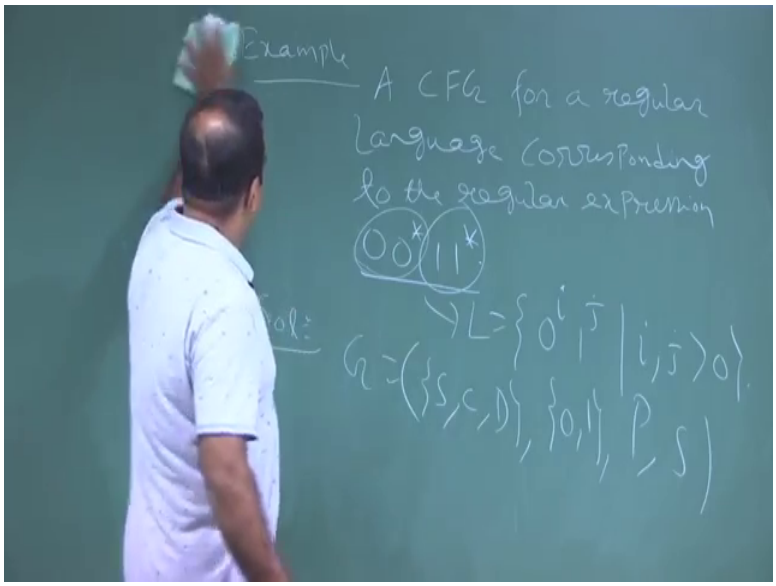


So, suppose we take a find the context free grammar; find a context free grammar, which is generating the following language, generating the language of even length palindromes even length palindromes.

So; that means, L will of this form w w R and w is belongs to this sigma is a b. So, now, how to generate it? So, we can take just S G to be the only we can start with only 1 variable S and then the alphabet terminals are just a b and P, S. So, what are the rules?

So, P is defined as S is going to a S a or b S a sorry bSb or epsilon that is all. So, there are 3 productions are there in P 1 is S S this is a S a b S b or epsilon ok. And now we can easily verify that this is accepting this language w w R the reverse of that. So, now, we can take some more example some more example of this.

(Refer Slide Time: 18:35)



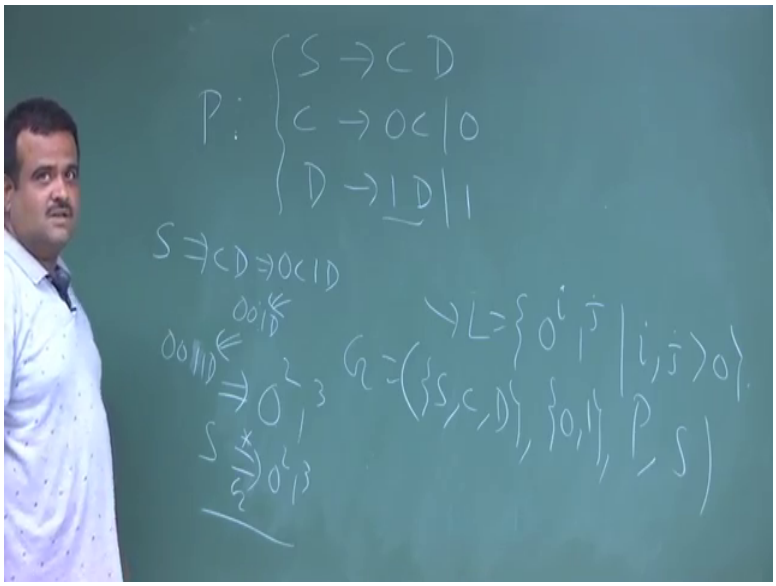
So, we want to have a context free grammar for the regular language for a regular language, which is corresponding to the regular expression; which is corresponding to the regular expressions 0 0 star 1 1 star.

So, this is our language. So, this is the regular expression. So, what is the corresponding language? It is 0 followed by any number of 0's, then followed by any number of 1's. So, this is corresponding to the language 0 to the power i and 1 to the power J or any i J greater than 0, because this 0 is there 1 is there so, i J must be greater than equal to 1 so, any number of 0.

So, we have to this is we know this is regular, because this is we have a regular expression for this. So, this is regular we do have a finite automata which can accept this language. Now, we want to find a context free grammar which can generate this language. So, any idea, how to do? So, it consists of 2 2 part 2 language one is this language any number of 0's and then followed by any number of 1's.

So, it is a 2 part. So, what is the solution? So, the idea is the language is concatenation of the 2 language; one is this one, another one is this one. So, that is why we will from this grammar like this. So, we will start from a S,C,D C,D we will be using for having the 2 different language and the terminals are the input alphabets P S. Now, what is the rules so, what is the rules? Rules are basically like this S is going to C D.

(Refer Slide Time: 21:21)

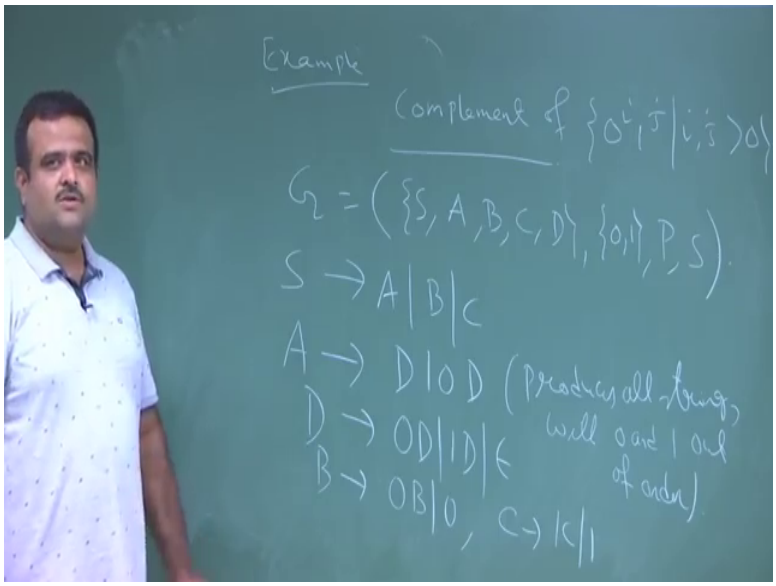


We use C for representing this 0's and D for representings the 1's. So, now, C is going to 0 C or 0 and D is going to 1 D or 1, that is all. So, this is our P, this is our productions, so, production consists of this. So, if you take any string of this form, which can be which is the sentential form of this grammar, any string of this form. Say 0 to the power 2 and 1 to the power 3 how to get this? So, we will start with a S C D, now C is going to 0 C and D is going to 1 D.

Now, this is again we need one. So, that is also this C is going to 0 0 and 1 D. Now, again D is going to this we have to have one more 1. So, we can have we can again apply this. So, 0 0 1 sorry 1 1 D, now this is going to here. So, this is reachable for this is this can be derive from this S. So, this is the sentential form say sentential form.

So, any form of this 0 to the power i and 1 to the power J will be the corresponding language ok. Now, we want to complement this, we want to get a context P grammar, which can compliment this. So, you have to compliment of this language 0 to the power i 1 to the power J.

(Refer Slide Time: 23:27)



So, compliment means? So, it will be not of the form 0's then followed by 1's. So, there is no obvious way to find the compliment in the context free grammar. So, we have to do some trial and error method.

So, there are 3 cases one case the string somewhere it is 0 again followed by one like this. So, this is the 0 1 in out of order, that is one possibilities and other possibilities is we have only 0's which string and other possibilities is we have only one strings. So, that are complement. So, if you do that. So, we want to find the complement of this. So, this will be like this, S we need few variables for those A B C D 0 1 P S ok.

Now, we need to defined the rules. So, that we can reach to the compliment of we can generate the compliment of this language ok. S is either A B or C. Now, we are going to use this A B C for different different purpose. So, a we are going to use for this all the string of 0 1, which is out of order; out of order means it is not that 0 followed by 1.

So, for that we need to take D a is going to D, 1 0 D. So, this produce produces all strings with 0 and 1 out of order. And D is going to 0 D or 1 D or epsilon. So, it is either 0 string or one strings, but we have 0 1 over here. So, that is the gesturing that property.

And, B is going to 0 B or 0, this is the 0 string and C is going to 1 C 1. So, this is the rules this is the production by which we can get the we can generate the language, which is the complement of this language.

Thank you.