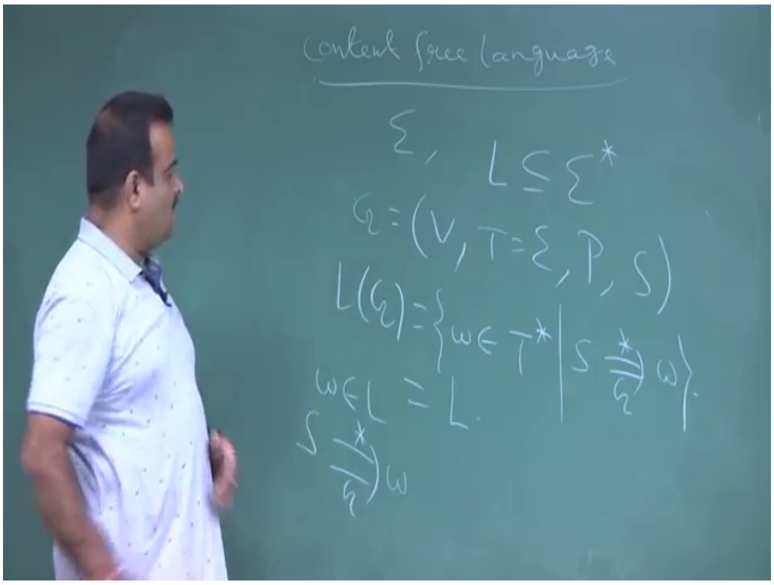
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 38
More Example on CFL

Ok, so we are taking example of the context free grammar; context free language. So, you are saying the if a language is a context free language or not.

(Refer Slide Time: 00:26)

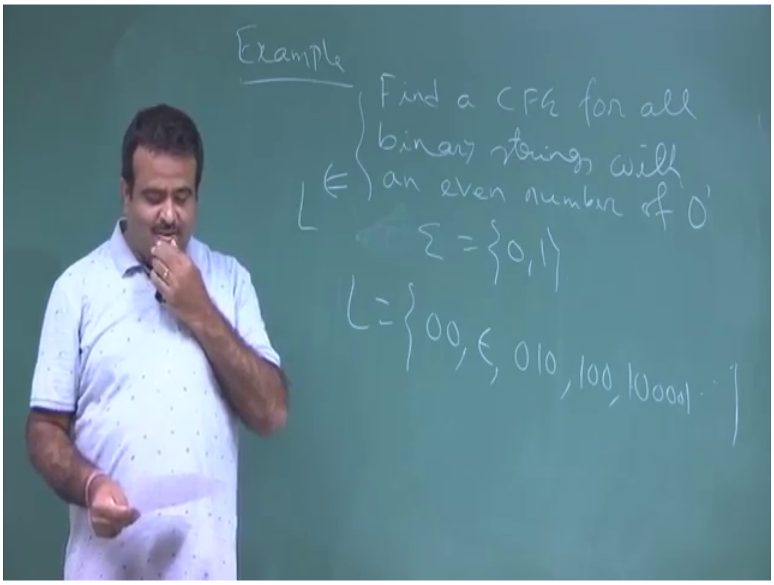


So, basically we have given a terminals or it could be 0, 1, it could be a, b, it could be a, b, c, 0, 1, 2 so anything. So, you have input alphabet which is also called terminal in that sense of context free grammar and we have a language L which is a subset of sigma star which is consists of the strings of coming from the alphabet of sigma.

And then we know this we call this the context free grammar, context free language if there exists a grammar G, T is here, P, S; S is the start variable such that this S is so, L is nothing, but L G is, but language generated by G which is the set of all strings of this terminals or the alphabet such that this is derivable from S by the productions.

Now, if this L is same as this. If the any string is in L we can derive it from S then it is called a context free grammar ok. So, there may be more grammar to generate a context free language. So, we will take an example of that; we will take an example of that.

(Refer Slide Time: 02:27)

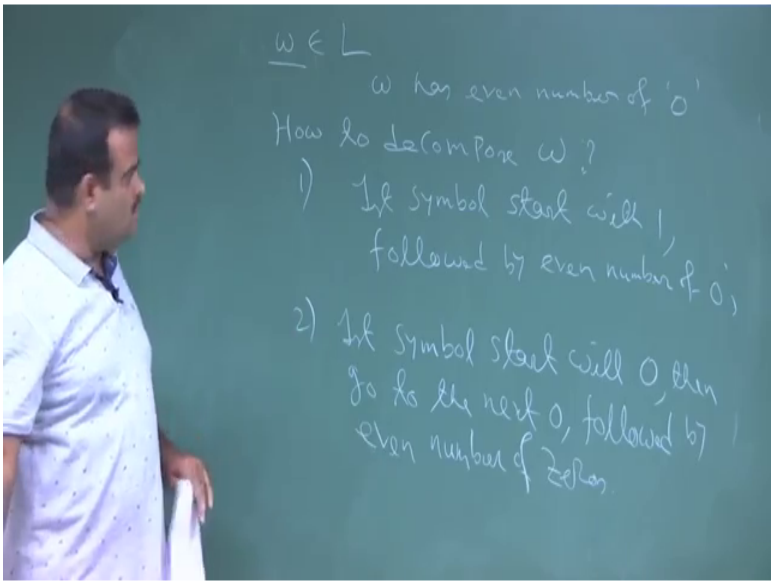


So, find a context free grammar for all binary strings with an even number of 0s; even number of zeroes. So, we just take a string of 0 1.

So, here T is I mean sigma is 0 1 input alphabet is binary string binary alphabet. So, we take a string and we count the number of 0s. So, number of 0s must be even. Say for example, if this is the language L so L consist of also this is 00 even number of 0 say epsilon we can say, no 0s 0 number of 0s, then 010, 100 like this so even 4 0s any position of this like this. So, even number of 0s. So, that is our L.

So, now we are want to see whether this L is a context free language or not. So, that means, we can we construct a grammar which can generate this L. So, basically we want to decompose the string, the string we take a string which is given number of 0s. So, how to decompose it?

(Refer Slide Time: 04:29)

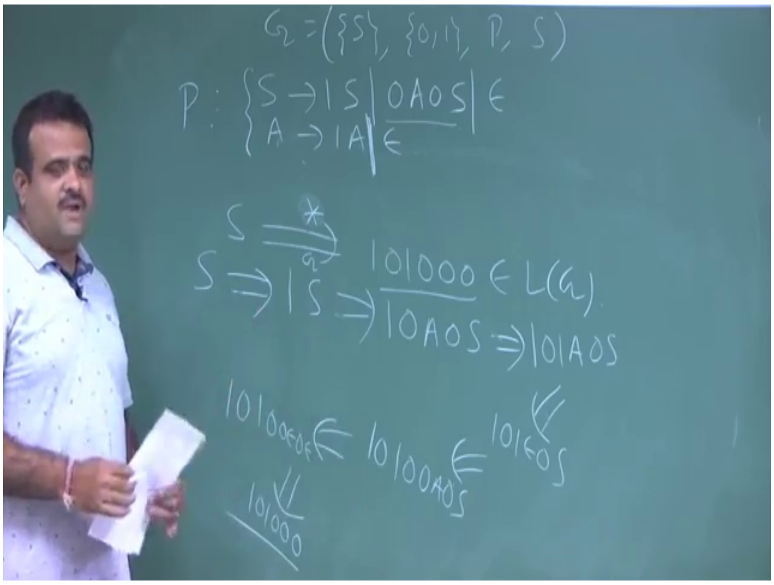


So, we so, this is belongs to L so where W has even number of 0s.

Now, how to decompose W? If you know then we can get the grammar we can get the production how to decompose W ok. So, that so, first one way is first symbol start with 1; first symbol start with 1 and followed by even number of 0s; followed by even number of 0s and second one is first symbol start with 0 then followed by odd number of 0s; so first symbol start with 0.

Since we have already have one 0, then the remaining will be have odd number of 0s, 0 then go to the next 0; then go to the next 0 followed by even number of 0s; even number of 0s ok. So, if we can see this decompose then we can write the we can construct the grammer how? So, let us try to construct the grammar based on these two decompose.

(Refer Slide Time: 07:03)



So, we have this grammar G. So, let us have only one symbol, one variable and then we have 0, 1 is the terminal or the alphabet and the production rule and S. So what is the production rule? Production rule is like this. So, S is going to so starting with 1 S or 0 then A 0 S epsilon. So, this is the first case because S is the S should generate the even number of 0s. So, if we have a starting it with 1 then still S to generate the even number of 0s.

Now, if we have a 0 then we will have to go for the next 0. So, this A is going to A is going to 1 A or epsilon 1 A or epsilon. So, this is our P; the productions; so this is our P. So, now, this by this decomposition we can get the we have productions; so that is the trick.

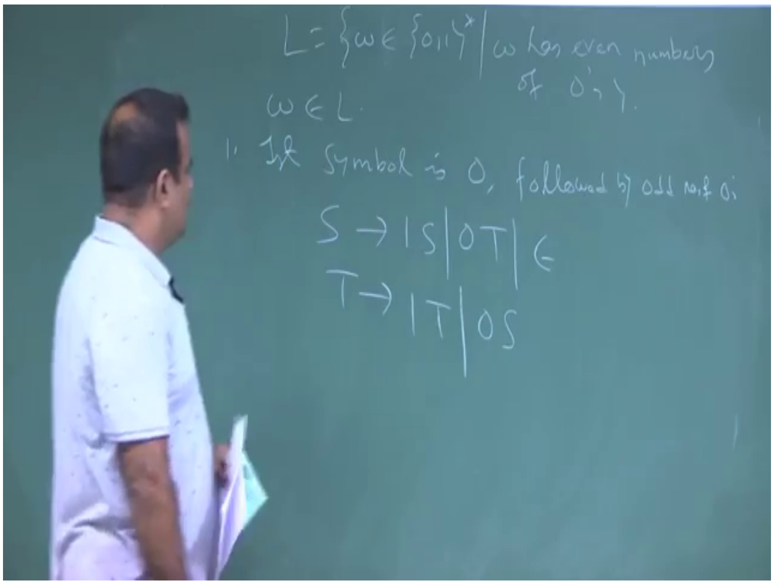
So, now we can verify this, this is giving has the same language. So, we can check whether say we take any arbitrary string 101000. So, this is how being how many? This is having 1 2 3 4 even number of zero. So, we want to check whether this is accept this is generating by this language.

So, how to check that? So we start with S, we derive; so we have a 1 over here so we need to follow we need to take this, then we can take this S to be this rule because we already have a 0. So, 1 0 A 0 S; now this A must go to 1 A because this 1 is there; so, 1 0 1 A 0 S ok. Now this now this there is no more 1 over here so now, this A must go to epsilon so this is 1 0 1 epsilon 0 S.

So, now, how many 0s we have after 1. So, we have one 0. So, we still need to have two more 0. So, we will take help of this 1 again. So, 1 0 1 0 0 A 0 S so we already have three 0s now we can make this is epsilon 1 0 1 0 0 epsilon 0 and in the next step we can make S also epsilon. So, this is nothing, but 1 0 1 0 0 0 this one. So, this can be denied from S how many step 1, 2, 3, 4, 5, 6, 7, 8 actually so this is 8, but we can make it star like after; after few derivation we can reach to that string so this is accepted. Now this way we can verify that any such string is accepted by this; however, even number of 0s.

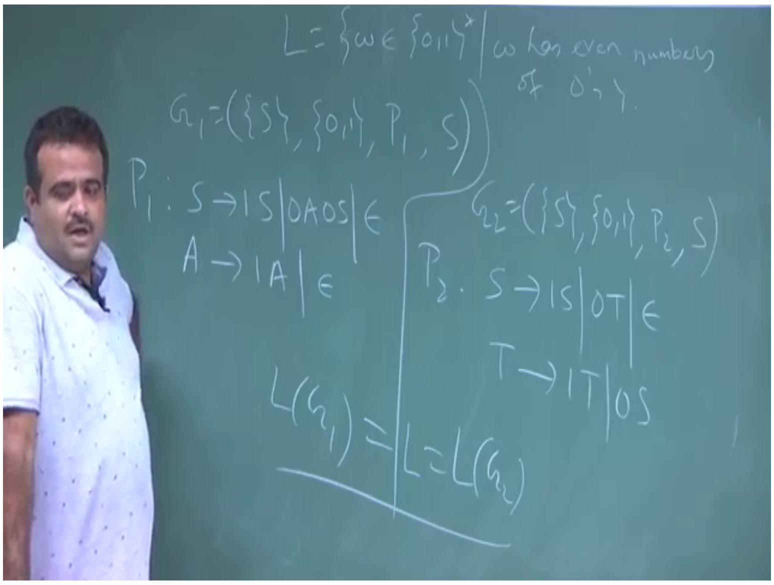
Now, we can have another grammar for this ok. So, we can have another grammar for this. So, for that we need to decompose like this yeah.

(Refer Slide Time: 11:26)



So, the idea is so this L is at the set of all W such that W has; W has even number of 0s. Now if you take a W from this, now earlier we decompose now we can decompose in another way like this so first we generated first symbol is 0, then followed by odd number of 0s; followed by odd number of 0s like this. So, we can have like this so S is going to 1 S 0 T this is this one or epsilon this is this one or second one is it can start with even; it can start with symbol 1 and then followed by again even number of 0s and then T is going to 1 T or 0 S.

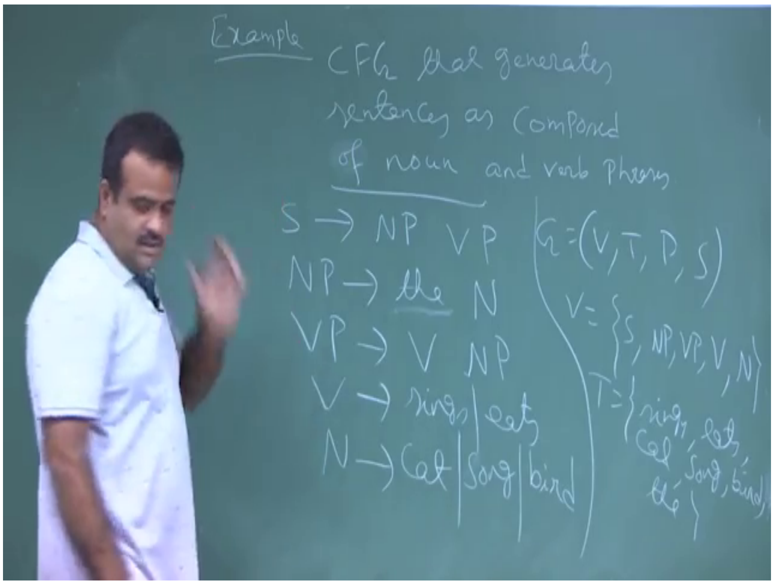
(Refer Slide Time: 13:13)



So, let me just write this so, let me just write this grammar. So, our G 1 the grammar 1 is earlier grammar. So, v is same only one symbol, this is 0 1, P 1 and S is. So, what is P 1? P 1 is the earlier one; earlier one was S is going to 1 S or 0 S or epsilon and A is going to 1 A or epsilon so this is our 1 G and another G 2 is you take some S, 0 1 P 2, S so we can change the S also it does not matter. So, what is P 2? P 2 is S is going to 1 S, 0 T or epsilon and T is going to yes that only 1 T or 0 S ok. If we have a 0 then we must have a S because that will I mean we need to have even number of 0s. So, these two are different grammar, but they are accepting the same language.

So, L of G 1 is equal to L is equal to L of G 2. So, this is one example where we can have a more than one grammar for a given language. So, now, we take another example where we will come to know why this is called the context free; why this is called context free grammar or context free language I mean so to understand that let us take one more example.

(Refer Slide Time: 15:26)



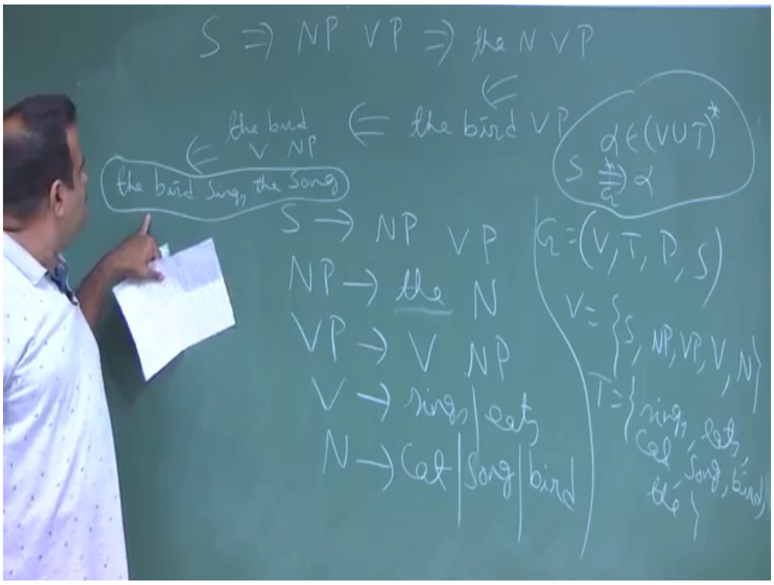
So, we are looking for a grammar, context free grammar that generate sentences as composed by composed of noun and verb phases; phases; phases ok.

So, we are looking for a grammar which is; which is give us this English sentence which are composed by noun and verbs. So, like the bird sings the song like this ok. So, how to do that? So, let us draw a so S is going to say noun phase, then verb phase. Noun phase where it can go the and noun and the verb phase verb then the noun phrase and the verb is say there are say two verbs sings and eats. So, sings or eats and there are nouns are say cat, song and bird; cat song and bird ok. So, now we will see what are the grammar, what are the string we can accept we can generate out of this grammar ok.

So, this is a grammar for this sentence I mean the sentence has composed of nouns and verb phases. So, this is noun phase, this verb phase. So, what is the G is so this is our P G is V, P V, T, P, S. What is V? V here is S noun phase, verb phase, then the verb and the noun and what is T? T is the these terminals like sings, eats, then cat, song, birds this is our T the; the is also there and S is the starting screen.

Now we want to get some I mean we want to get some string of terminals which is generating by this grammar. So, let us try that ok.

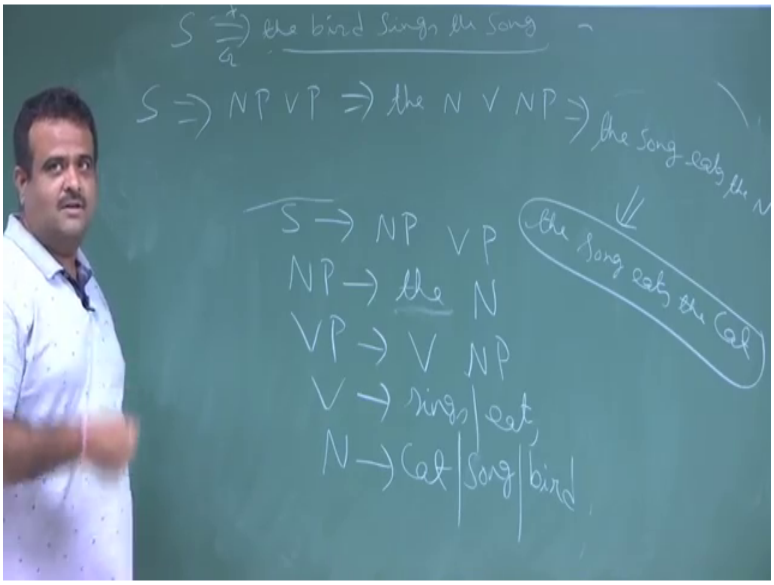
(Refer Slide Time: 19:30)



So, what will do we start with S, we take this S is only going to NP, VP, then say NP is going to this rule the N V P and say VP is going to VP is going to V N P again, VP is going to the N is say N can go any 1 of this so, I can write bird then VP.

So, VP is going to now V NP. So, the bird V NP. So, V can again go to; V can again go to either one of this so the bird sings and the NP, NP is the N; the N, N again go to say song. So, this is derived from this is a sentential of this. Sentential means which can derived from alpha is called a sentential which is a V cross terminal, if alpha is derived from S by the rule of by the production in G. So, this is; this is a sentential; that means, this is generating form this S; so S is generating this.

(Refer Slide Time: 21:37)

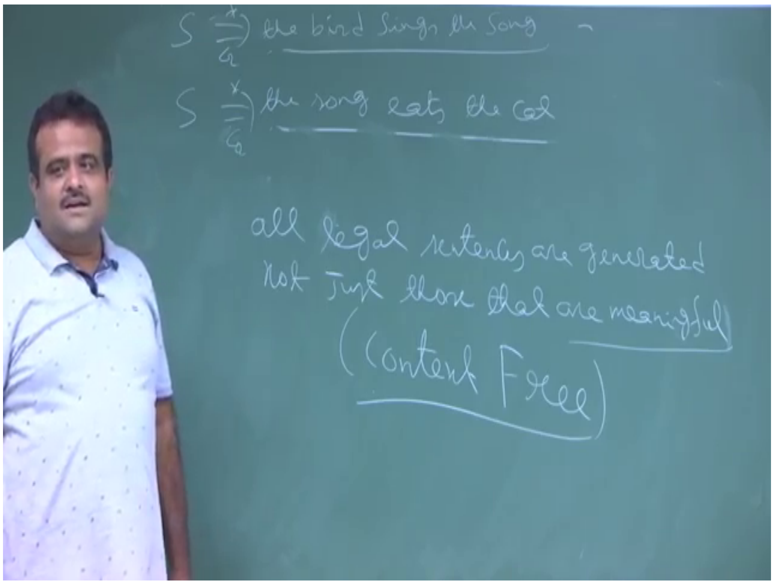


So, S so the bird sing the song. So, this is a; this is a string which is accepting which is generating by this language ok. Now we will try to get another string; we will try to get another string out of this.

So, so we start with S. So, from S we can go to NP VP this no other option NP VP and from NP we can go to the N VP. So, VP VP also you can go to V NP, now from here we can go N, N can go to any one of this; so N can go to song. So, the song and V we can go to any one of this eats, an NP; NP again go to the N.

So, now this N again go to any one of this. So, the song eats the N can go to [Laughter] cat. So, this is the terminals this is the there is no variables there. So, this is a; this is a string generating by this grammar and it has no meaning the song eats the cat. So, that is why it is called context free. We just follow the rules and we reach we do not I mean it may not get a meaningful text. So, you just follow the rules; so that is the sense that this things is called context free.

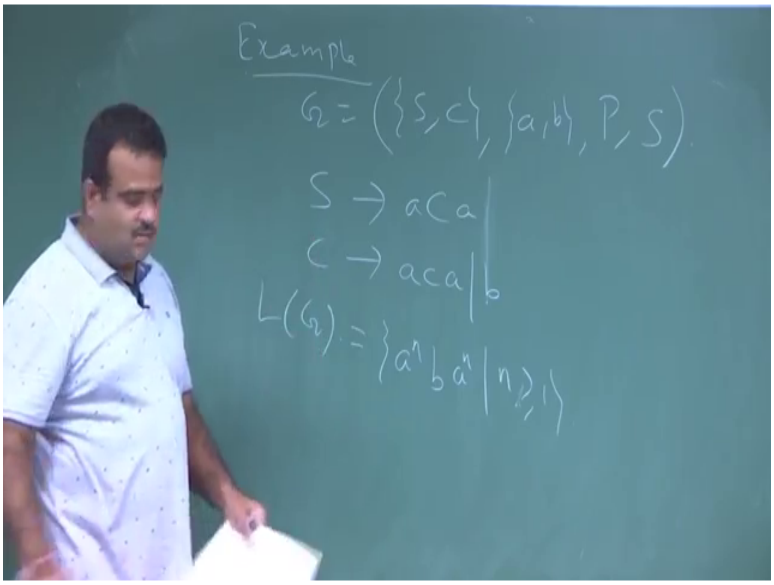
(Refer Slide Time: 23:56)



So, all the legal sense so, this is this is so that means, S is generating this also and S is this grammar is generating this and this grammar is generating this also the song eats the cat. So, it has no meaning, but it is a legal sentence we can get all the so all the legal sentence are generated not that those has a meaningful that is why it is called context free.

All legal means grammatically I mean English text legal sorry lel all legal sentences are generated or can be generated not just those that are meaningful; meaningful. So, that is why it is called context free. We just follow the rule, we just follow the grammar and we end up with all the possible sentences which it can form legal sentence all the few sentence has no meaning like this has no meaning in the English text, English text the song eats the cat ok; so this is the sense it is called context free.

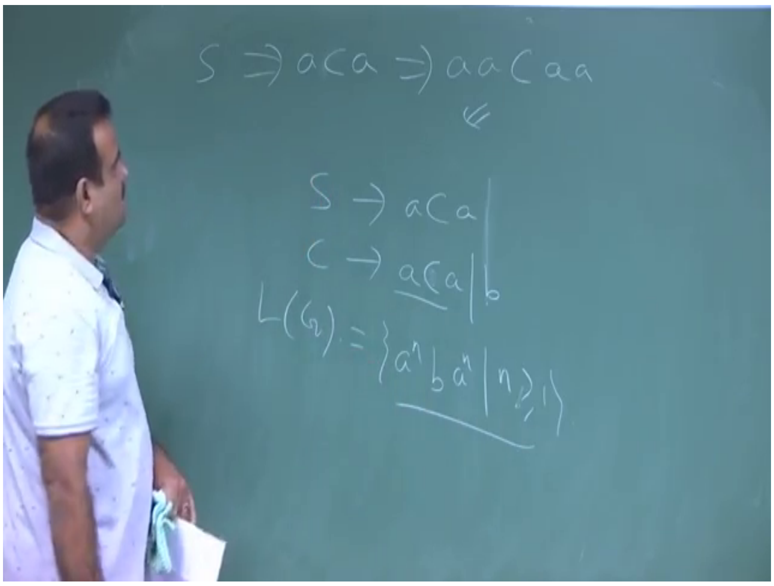
(Refer Slide Time: 26:07)



So, yeah so let us take one more example ok. So, just a quick example suppose you have a grammar like this. So, our we have two variable S and C and the alphabet or the terminals are a, b and P is the production and S. So, P consists of say S is going to aCa or S is going to S SCa and C is going to where C is going to aca or b.

Now, can you tell you what is the language corresponding to this what is a L of G? Now we can exercise this, the my claim is L of G will be a to the power n b a to the power n is greater than 1 ok.

(Refer Slide Time: 27:23)



Now from S we have to start with this so, from S we can; from S we can go to C no doubt there is only one options. Now, from c this is capital C from C if we go to b there is two option, if you go to b then it will give us ab aba or if we again apply this then it will it is capital C then it will be aacaa.

So, again if we apply this one b then it will give us a square b a square otherwise if we apply again this one so eventually it will be of this form ok. So, the a to the power n this will be belongs to this again if we take a W which is generating by this grammar we can prove that W will be up this from.

Thank you.