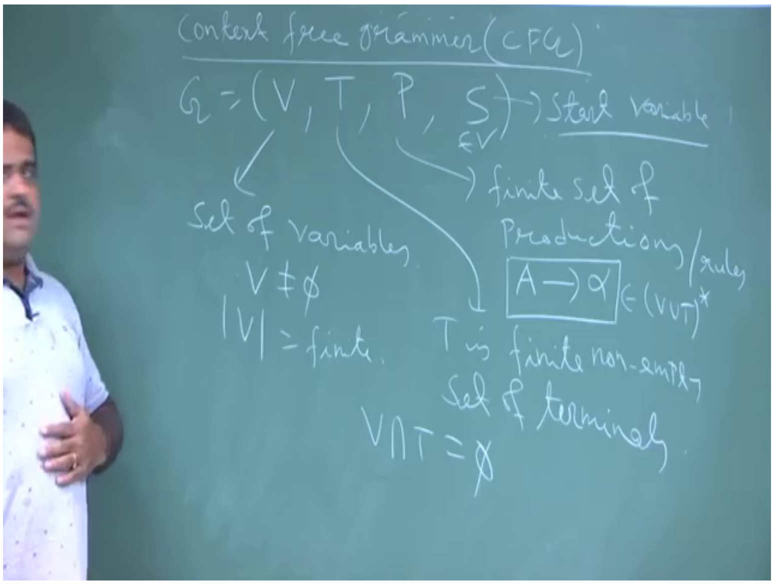
Introduction to Automata, Languages and Computation

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

Lecture – 36
Context Free Grammars (CFG)

So, we will start the Context Free Grammar, we will introduce the will define what we mean by context free grammar.

(Refer Slide Time: 00:29)



So, it is basically a 4 tuple. So, context free grammar is a 4 tuple which consists of V which is the set of variable, T set of terminals or P P is the set of propagation or set of rules. And this S S is a starting variable S belongs to V S is a starting variable. So, what is V? V it is a finite set of variable; V is the set of variable and this set is finite and it is non-empty.

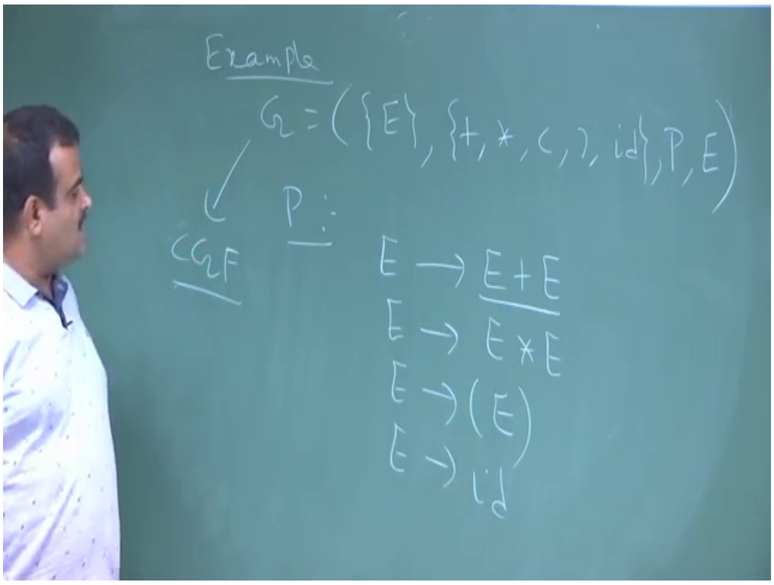
So, V is non-empty so V cannot be empty, but this is finite cardinality of this is finite. It is finite set of non-empty the non-empty set of variables. And what is T? T is the terminals again this set is finite; T is the non-empty finite non-empty set of terminals ok; that means, which elements are called terminals. And P and here we have V and T there should not be any intersection with V and T. I mean the terminal cannot be a variable and the variable cannot be a terminal so; that means, they are two disjoint set.

And the P P is the rules or the productions. This is also finite set of productions or rules ok. So, this P will be of the form say A is A variable A is going to alpha and this alpha alpha could be I mean any string between; so, alpha is coming from V union T star. So, alpha consists of including the variable any string consists of variable terminals anything. But here right hand side it should be a variable; right hand side it should be a variable A.

But, the left hand side A is producting to alpha or A is going to alpha kind of thing; A is this is the rule A is going to alpha. And alpha could be anything alpha put consists of BC then some terminals like this; we will take an example like this. But, if this alpha is belongs to alpha is a string consists of variable and the terminals. And alpha would be epsilon also that is why we have put it star over here. And S is the; S is a special variable which is called start variable, S is the start variable.

So, S is S belongs to V this belongs to V which is a start variable; this 5 tuple is called a context free grammar. So, we will take an example then it will be more clear; we will take an example of a context free grammar. So, let us take this 5, 4 tuple sorry 4 tuple.

(Refer Slide Time: 04:45)



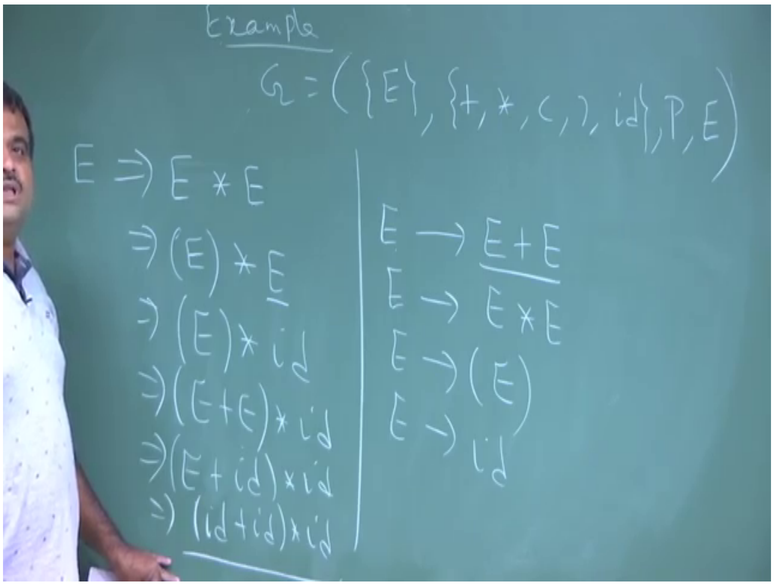
So, we have only one variable that is also starting variable and we have many terminals. So, plus star bracket these bracket then id these are all terminals. And then we have a rules and we have start because, there is all the one variable that is shall we say starting variable S ok. Now, we have to define the P the rules. So, rules we can have like this. So, E is going to E plus E. So, remember this all this belongs to V union T star.

So, this is a string of; this is a string of variable and the terminal both, but this side it will be a it must be a variable. So, that is the so, this is one rule. So, it consists a P consists of the rules. So, E star this star this star is nothing to do with our this star ok. So, this is symbol star; it could be multiplication in the arithmetic sense. This one and E is going to id these are the 4 rules we have ok.

So, this is the context being this is a context free grammar CGF: Context Free Grammar ok. And these are the productions rules, these are the productions E is this is the arrow means it is going to going means the production. E is producting a plus E E is producting yeah E star E is producting bracket E E is producting id. So, this is the production set and that is P ok.

Now, we will; now we will have some language which is we can have some derivation. What do you mean by derivation? So, suppose we have given this then we have to define the derivations like this. So, the derivation means we start with a variable and using the productions where we can go like this. So, if it is by one production rule we will apply this is a direct derivation otherwise it will be start like this.

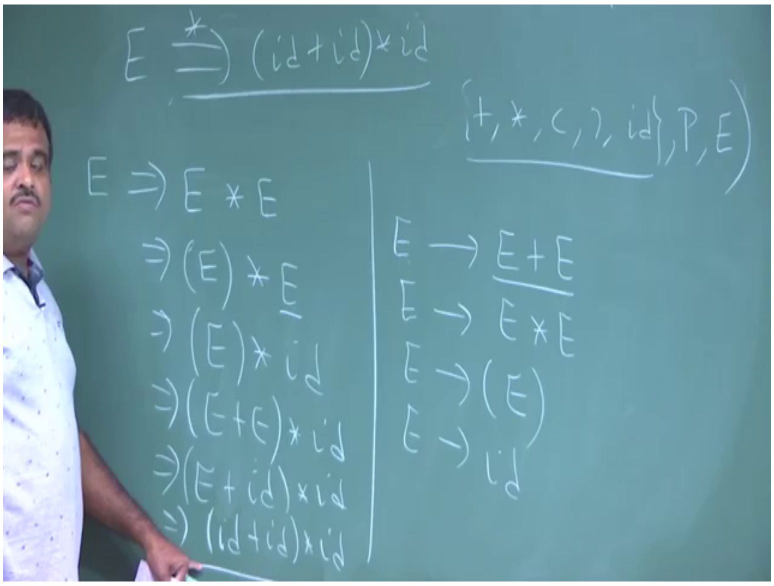
(Refer Slide Time: 07:56)



So, E E is so, let me just write this are the rule. Now, E is going to we can take any one of this say we take this production E star E ok. Now, these E star is going to again this E we can take any of this production E star E ok. Now, this E star is going to again this E we can take any of this production. Then again any one of this E we take this E using this production you can write this as id. Then again this E we can use this production, we can write this E plus E id ok. Then again we can use this E using this production we can write E plus id star id; then again we can write this using this id plus id star id.

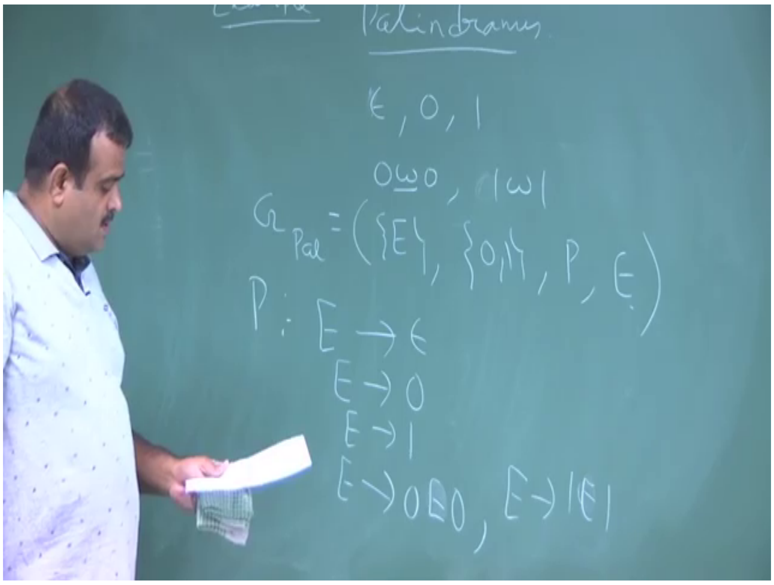
So, this all this there is no; there is no variable in this string, this string only consists of the terminals. So, eventually we can reach to we can derive this is the derivation. We can start from the starting variable and we can eventually reach to a string; we can reach to different strings. So, those are basically if we collect this those are called language of this grammar; we will formally define that. This is one of the string, string of terminals is this one. We have all these no variables, we do not want any variables in the final state ok. Final state is consist of all the string of terminals, so, this is called derivations. So now, this is one derivation.

(Refer Slide Time: 10:06)



Now, if you have multiple derivation we can say that star, I mean the multiple derivation we are reaching here id plus id bracket star id. This star means we are applying if it is direct one derivation then we can just use this derivation symbol otherwise if it is star. So, this is so this is the final one we want, no variable only the terminals. And those will be collecting those string of terminal will be our language accepted by this context free grammar. We will formally define that.

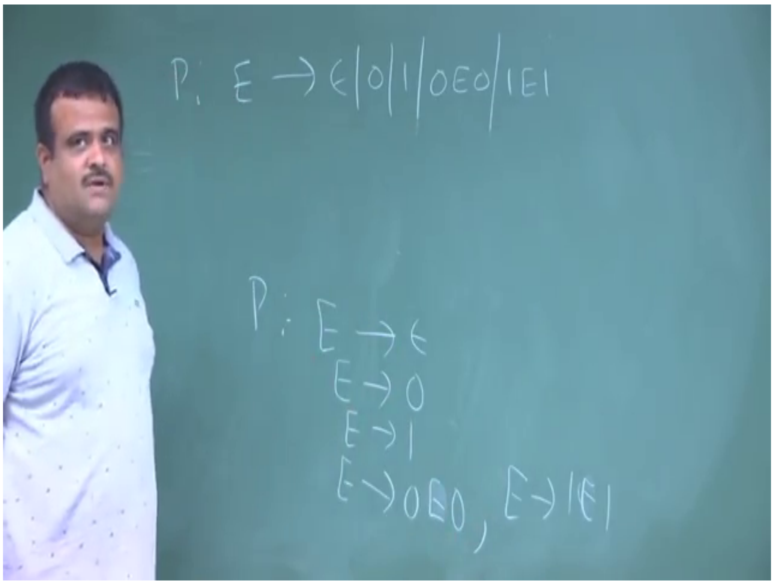
(Refer Slide Time: 10:57)



So, let us take another example say palindrome strings. Palindrome means if we read I mean reverse is also same as the forward string. Like this string 1 epsilon 0 1 these are all palindromes. Now, if 0 w 0 is a palindrome where, w is palindrome was 1 w 1 is a palindrome ok. Now, we want to have a context free grammar which is accepting the palindromes like in the final terminal string will be the palindrome. So, that we want to do that; so, that is a G palindrome.

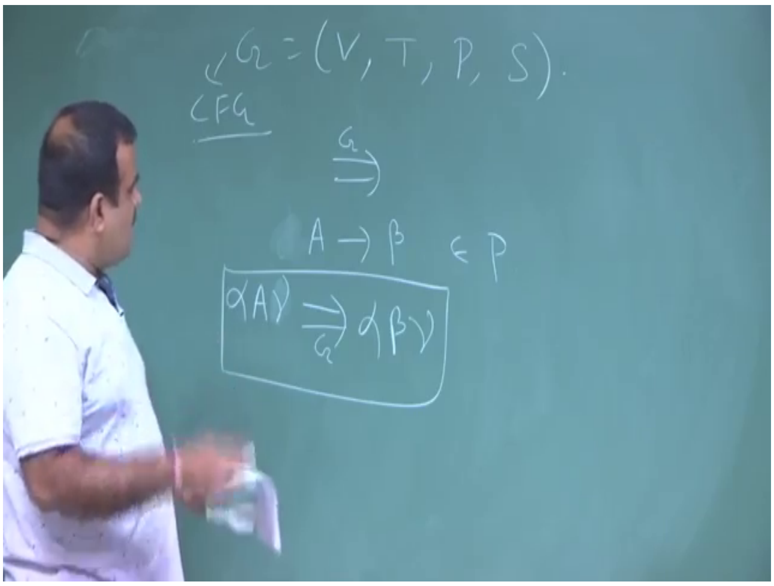
So, we can have a starting state because I have only one state then the 0 1 the rule and S S is on the one state. Now, the rules will be from this we can observe because these are all palindrome. So, E can go to epsilon or E can go to 0 or E can go to 1. These are the direct options or E can go to 0 w 0 or E can go to 1 w I sorry 1 E 1 E 1 ok. So, this is the way we define the rules of the palindrome grammar. So now, this there is another way to write these productions. So, we have this E is going to many things, E is going to epsilon, E is going to 0, E is going to this.

(Refer Slide Time: 13:28)



So, this we can combine and write in the same line like this E is going to epsilon or 0 or 1 or 0 E 0 1 E 1 like this. So, just it is a in a one line we can write this because, all the variables are E. So, E is going to either epsilon first rule, E is going to 0 second rule, E is going to 1 third rule, E is going to E 0 E 0 fourth rule, E is going to 1 E 1 P through. This just another way of writing this nothing else and this is accepting all the palindromes ok. So now, we will formally define the language accepting by a context free grammar.

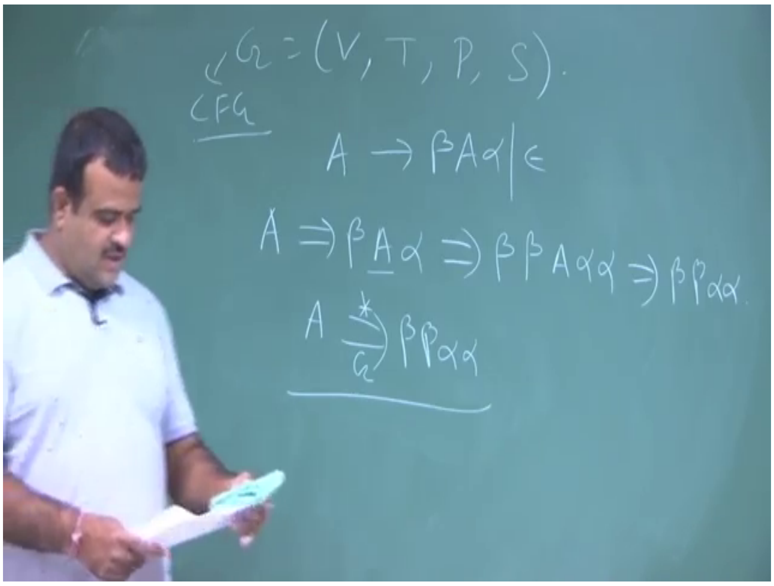
(Refer Slide Time: 14:28)



So, suppose you have a context free grammar G which is V T P and S ok. So, this is our CFG; now we already have this symbol the derivation. Derivation means we have like a if there is a rule in this P. So, suppose A is going to beta, this belongs to say P this is a rule ok. Then if we have alpha A beta this we can say this is going to alpha A is going to beta sorry, this is gamma alpha A gamma this under G.

This is the derivation because we have a; we have a string A alpha A beta A alpha A gamma ok, alpha gamma could be anything any it could be variable. Now, we have a rule in P where A is going to beta so, that direct rule we can apply here A is going to beta and that is the derivation. So, this string is going to this string this thing is derived from this string by direct application of this ok. So, that is the meaning of this.

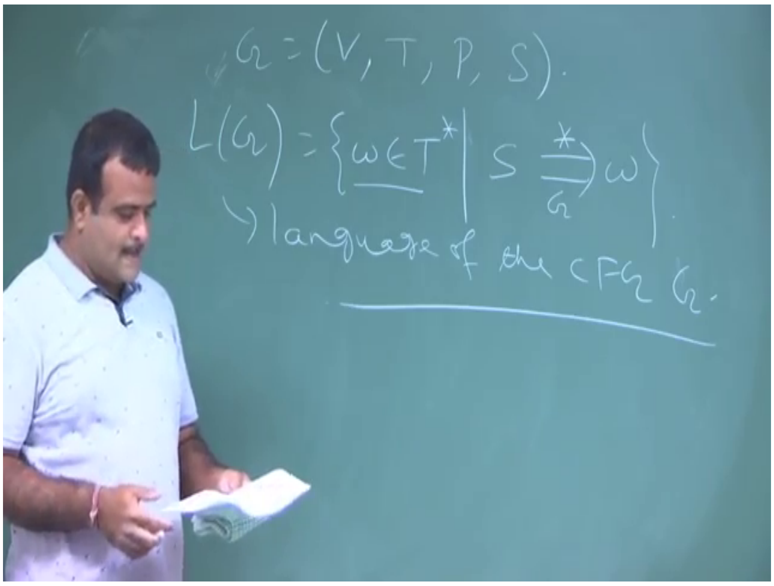
(Refer Slide Time: 16:15)



Now, if you repeatedly apply this like again if say like this. Suppose we have say A is going to beta A alpha or A is going to epsilon. So, what we can say? So, we can just start writing deriving. So, A is going to beta A alpha. So, again we can say this A is going to so, A A is going to beta A alpha. So, beta A alpha alpha; now we can just write A is going to alpha or we can take more, so, beta beta alpha alpha.

So, basically A is this is derived from A, but not by the single rule we have applied the multiple rule. So, that is why we can write this is star beta beta alpha alpha. So, this is derived from A by applying the more the one rules that is why it is star ok. So, that is the just the notations derive derivation notations ok. So, now, we will define the language accepted by this grammar G.

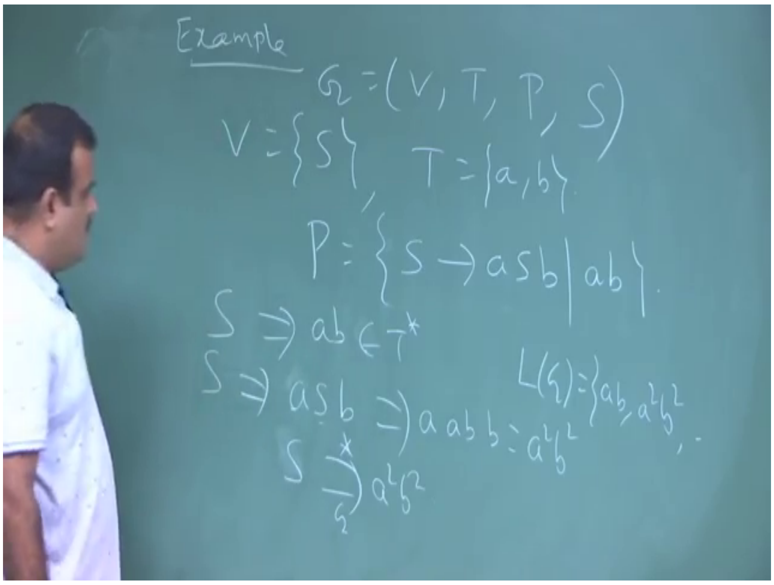
(Refer Slide Time: 17:48)



So, that is denoted by L of G. So, this is the set of all string of terminals, in the language we do not have string of variables. So, this is only the string of terminals; string of terminals such that we have from S we can derive w by applying the repeated I mean multiple rules using the rules of G. So, there are two things are there so, this has to be at this w the string is a terminal. So, this language is consists of the string of many time a string of terminals only.

And this terminal should be reached which means this terminal should be the string of terminals would be derived from the starting variable S. Then this set of this set of string, if we gather this set of string is called the language of the context free grammar and this is. So, this is the language of the context free grammar language of the context free grammar G, this is L of G ok. Now, when we say given a so, we can take some example before that we go further.

(Refer Slide Time: 19:38)

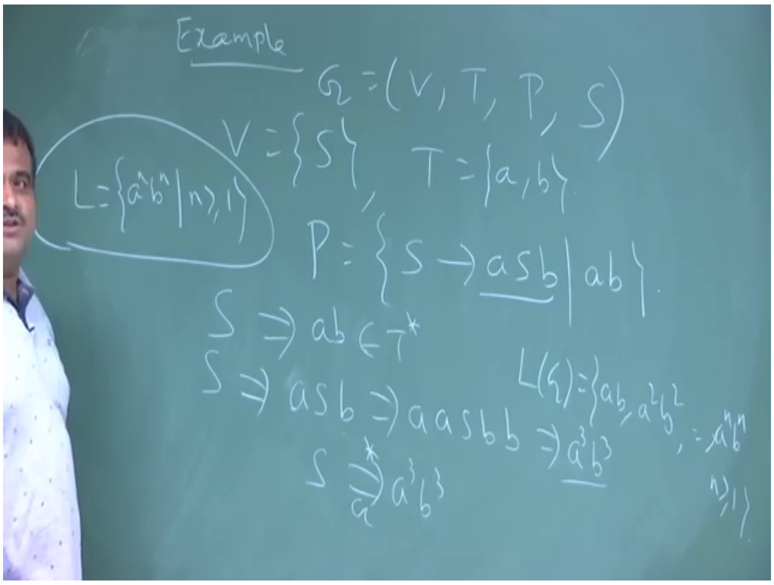


So, suppose we have we defined this V T P S where, V is only one symbol S and T is terminus which is consists of a b. I will just say input alphabet which you have seen and the P P consists of the rule rules, if say S is going to a b sorry. S is going to a S b or S is going to ab this is a context free grammar. Now, we want to find the language accepted by this grammar. So, what are the language accepted by this grammar?

So, it is it will be the all string of terminals where we can reach form or which can be derived from the starting variable S. So, you start with S so, we can just take either one of the rule; so, a b can be derived. So, a b belongs to a b is the; a b is the string of terminals. So, this L of G this consists of a b then what is the next. So, from S we can apply this rule S sorry a S b then we can apply a b. So, a a b b so, this is a square b square this is also set of terminals; like this is also string of terminals. So, this is a square b square ok.

So, similarly we can take so, that that means, this a square b square can be derived from S using the rule of G ok. Now what about the next? Next is say we can instead of applying a b we can keep on apply this.

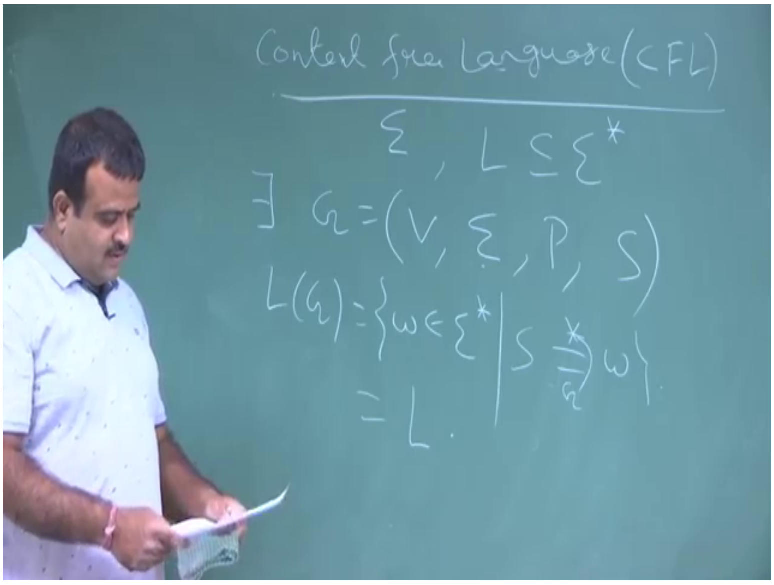
(Refer Slide Time: 21:56)



So, we can apply again this rule the first rule first production. So, again S is going to a a S b b. So, again we can apply S is going to a b then it will give us a cube b cube. So, that means, a cube b cube it is derived from this S by applying the rules of G. So, S a cube b cube; so, in general it is a n b n here, n is greater than equal to 1.

So, this is the language you remember this language, if you just a to the power n b to the power n greater than 1. This language we have seen earlier and here proof that this language was not a this language is not a regular language, but this language is a context free language. That means, we have a context free grammar which is accepting this language. So, anyway we will talk will define the context free language.

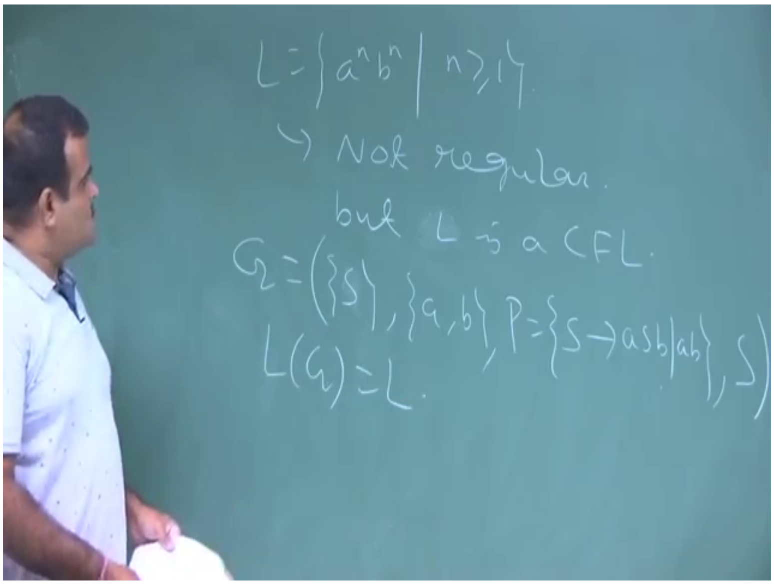
(Refer Slide Time: 23:24)



So, let me define the context free languages CFL. So, we have given a input alphabet. So, that is a finite set of alphabets. Now, we say this is a this we have a subset of star. We say this is a context free language, if there exists a context free grammar G which is V T is nothing, but sigma over here and the rules and some S such that this L is such that this language of G which is nothing, but this string of terminals.

Here terminals is the sigma such that S can be I mean this w can be derived from S using the rule of G and this will be L. So, if we have a context free grammar from which each of the string can be derived from that starting variable then we say this is a this collection is a context free language. So, which will corresponding to this context free grammar and this is the language of the context free grammar ok. So, that means, just now we have seen the language like a to the power n b to the power n.

(Refer Slide Time: 25:36)



So, a to the power n b to the power n where, n is greater than 0 greater than 1 ok. Now, this is this we know this is not regular, but this is a context free language. But, L is a context free language. Why? Because, just now we have seen you have a we can have a context free grammar. So, V we can have just only one variable S and then we have sigma a b or T we can say terminals.

And the P P is nothing, but S is going to a S b or a b. These two rules and we have S and see L of G is nothing, but L ok. So, this is a context free language, but it is not a regular language ok. We will talk more on this.

Thank you.