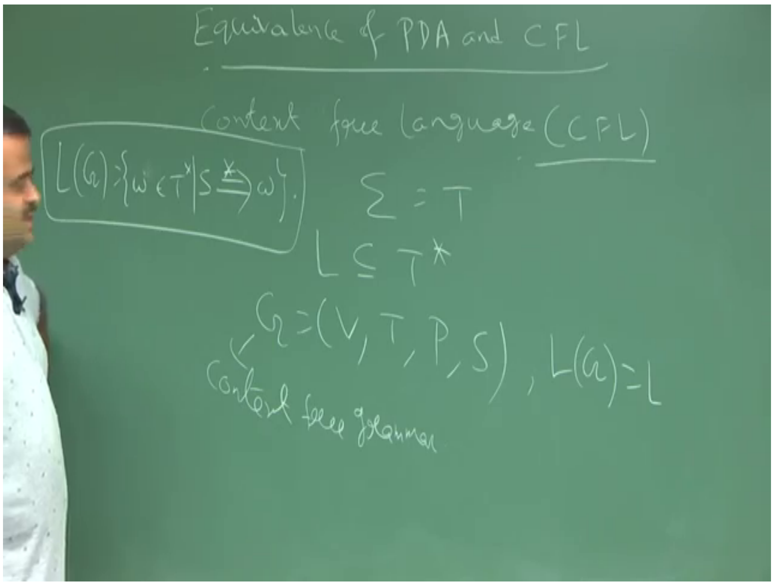
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 54
Equivalence PDA and CFL

(Refer Slide Time: 00:15)



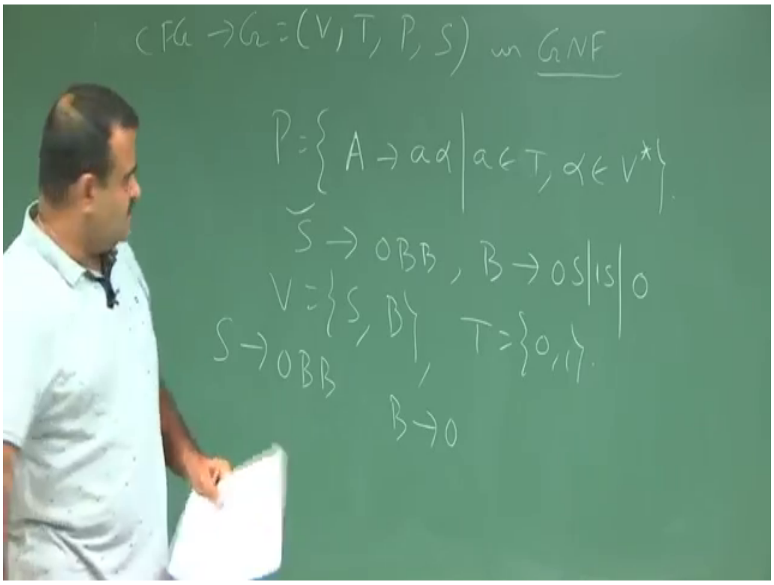
So, we will talk about the equivalency between the PDA and CFL: Context Free Language. So, we will see if your language is actually generated by a context free grammar, then we call that language is a context free language. We will see that this is they are equivalent like a given the context free grammar I mean given the context free language we can generate a feed; that means, they will acts that will be generated by a context free language a context free grammar.

And, then we can construct the corresponding PDA which will accept the same language acceptance may be in either by acceptance by stack empty stack we will see by empty stack, but both are equivalent if it is empty stack then we know how to convert that to the accepted by the final state. So, let us talk about this context free language, context free language CFL. So, given a sigma which is the input alphabet or terminal sometimes we say this is the terminals in the sense of context free grammar. So, what is the context, what is a context free language?

It is a string of terminals or string of alphabets input alphabets such that we have a grammar G which is V T P S this is a context free grammar such that it will generate the language L ok. So, what is L of G? L of G is nothing, but we start with set of all string or terminals sorry all string of terminal such that we state we start from S and we should able to the derivation the derivation. We should able to reach to w this is called our the grammar generated by this context free generate this grammar generated by this CFL ok

And you should given a given a grammar if we have such a I given a language if we have such a grammar then that language is called context free language. So, now, we will come to know the equivalency between the context free language and the PDA. Given the context free language we can construct a PDA which will accept the same grammar so that we will see. And, for that we need to take this grammar in a particular form which is Greibach Normal Form GNF, even if it is not in the GNF we can construct that also we will see, but first let us take this grammar in GNF and then we will try to construct a corresponding PDA which will accept the same language.

(Refer Slide Time: 03:51)



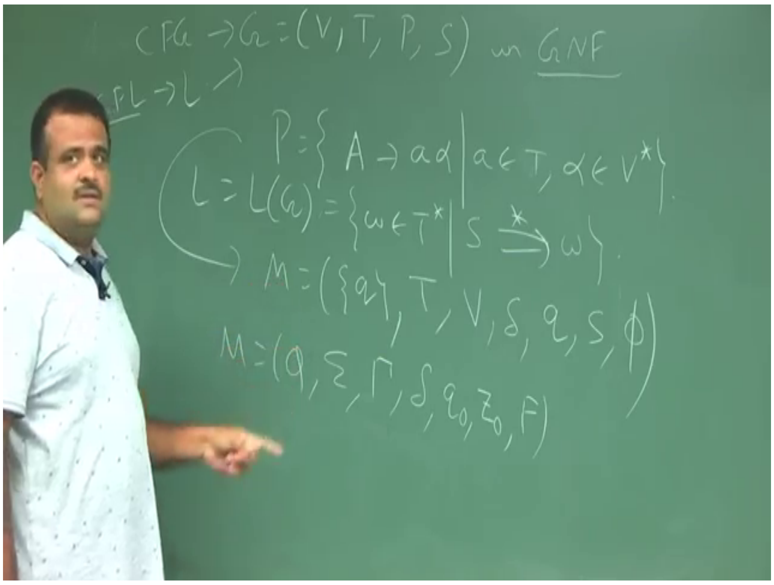
So, let us take a grammar G which is basically V T P S they this is a let GBA context free grammar in GNF in Greibach Normal Form. So, Greibach number form means the all the products we know the two normal form right one is CNF another one is GNF ok.

So, what is GNF? GNF is nothing, but all this productions are of this form A is going to a alpha where a is a terminals, and alpha is a string of variable then since it is star. So, that means, alpha could be epsilon also then if all the productions of this form, then we call this grammar is in GNF: Greibach normal form.

So, for example, we have seen some example of a Greibach normal from like if we S is going to say 0 OBB and B is going to 0 S 1 S or 0. So, if you see here V is nothing, but we have two variables terminals are 0 and 1 and the productions are of this form there are. So, all the production like if S is this productions if we take. So, this is of this form that a some symbol is going some variable is going to alpha this is alpha some terminal alpha, this all this like this even B is going to 0, this also because this alpha could be epsilon because this is the coming from V star. So, string this string would be null string also ok.

String of length 0 null string ok; so, we have given this; we have given this and all these are of the form in the Greibach normal form. Now from here so, how to suppose it is generating the, suppose there is the a CFL L for which we have this that that means L is equal to L of G.

(Refer Slide Time: 06:31)



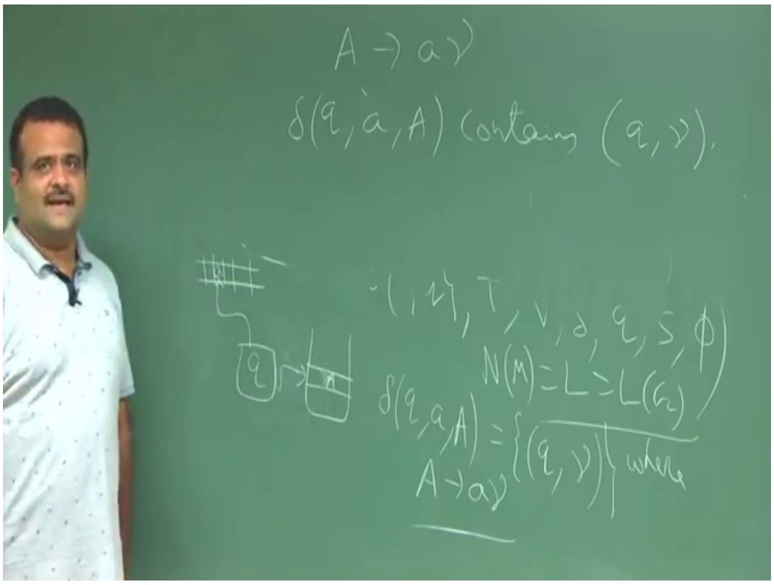
L of G is nothing, but set of all string such that S is derived derivation three if you recall leftmost derivation and rightmost derivation. So, here we will use the leftmost derivation to prove it. So, anyway let us see that. So, this is going to this way ok. Now, we want to generate a grammar a PDF, a PDA Push Down Automata which will accept the same L.

So, from these we want to generate a PDA which is Q sigma. So, sigma will be T here ok.

So, q is we are only take the only one single state. So, this will be Q only one stated this is one state PDF and sigma is this and v the v the said if you recall the PDF is of this from Q sigma then this is the symbol stack symbol. So, stack symbol you are going to use the variables as a stack symbol. So, input the variable into the stack and then we have a delta; delta we have to define so that it should accept the same language; we have only one variables a only one state. So, Q is the starting state as well and if you recall this is delta q 0 was there z 0 z 0 is the starting symbol of the stack.

Now, here S is a special symbol, S it e will be using for the starting symbol of this stack, and then we have a set of final state, but here we are going to show this acceptance in terms of the empty stack. So, that is why we do not care about the final state. So, we will put this part to be empty we do not have any final state because we are going to show that, this language of this in the with respect to the empty stack is same as L ok. So, what we are going to show? We have to define this delta; we have to define this delta such that the language accepted by the empty stack is same as L which is L of G this we are going to show.

(Refer Slide Time: 09:11)

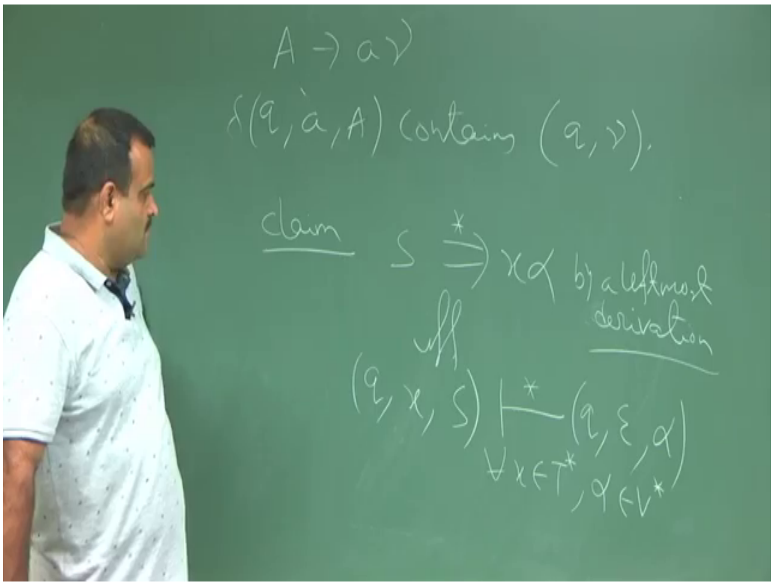


Now let us define delta. So, delta what is the input of delta? So, delta we will take some state q there is only one state q and it will read a input symbol say a ok. And it will use a stack symbol on the top of the I mean top of the stacks this is some z, this is the variable. So, this will be some A ok. So, delta is of this form q a; a is same as t which is sigma basically and we have a A and this will replace to what.

So, this is a non deterministic move. So, this will be going to yeah. So, this will be going to some this will contain this q and this A will move to gamma where we have a rule like this, A is going to a gamma where A is going to a gamma. So, this is the way we defined this. If A is going to a gamma is in this is in P then we have a this delta corresponding to that ok. So, now with this construction we can show that they will accept the same language.

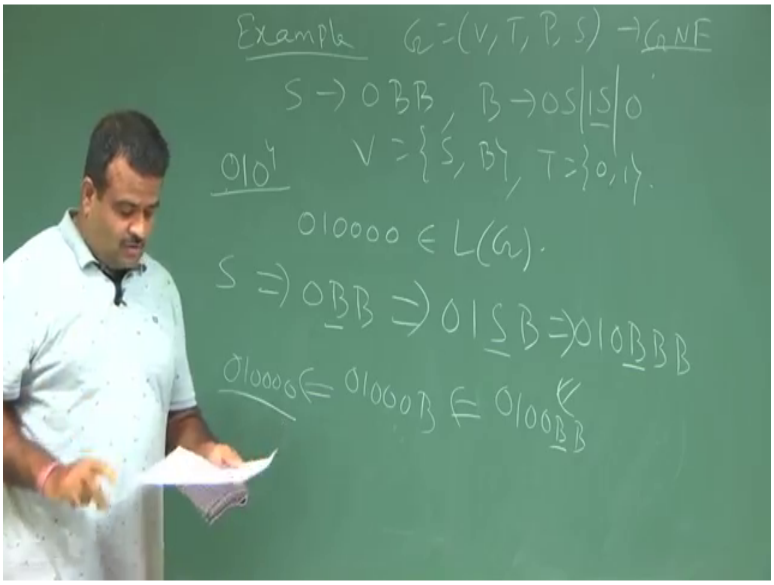
So, with this construction we can show they will accept the same language that means, whenever you have a rule like this A is going to a gamma, then we will have a then we will say q comma this; then we say that delta of q a A contains q comma gamma, q is the only one state. So, it will hop in that state, but only thing it should do some changes in the stack and slowly we have to empty the stack. So, this is the way. Now, it can be shown that if we do like this it can be shown that, our claim is it can be shown that.

(Refer Slide Time: 12:03)



S for the grammar if S is going to x alpha by the leftmost derivation if and only if and only if when a it means this is if this is happening, then this will happen if and only we start with q with a input x, and with the start symbol in the stack S this will reach to q, we do not care about the state alpha ok. This is true for all x and all alpha belongs to this stack. Now is alpha is empty, then this is the string of language generated by this grammar and also this is accepting by this q epsilon alpha. So, this prove is we are not I mean we can easily verify this prof ok. So, now, we will take a example to construct such a PDF.

(Refer Slide Time: 13:45)



So, let us take a grammar which is in GNF. So, suppose you take a grammar the one we have and B is going to 0 S 1 S and also 0 and what are the variables in we are using? So, this is a grammar V T and P S V is S comma B and T we have 0 1 and we have this S; S is the starting symbol and these are the rules.

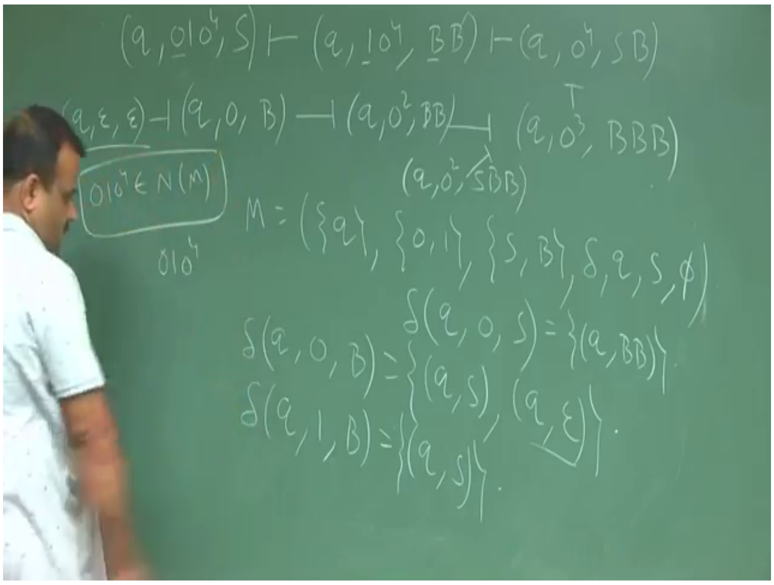
So, this is GNF ok. Now, we can see that this is a string 0 1 then four 0's this is a string accepted by this generated by this how we can generated this string? We can start with S, then we can use this rule 0 S, then we can use we can use the leftmost derivation because the way we are constructing is by using the leftmost derivation, then this S again we can there since there is a 1. So, you can again replace this by this rule. So, 0 1 S again then this S again it is sorry this not S this is B sorry I made a mistake. So, S is going to so this is not S. So, the S is going to 0 BB ok.

Now, since this is a leftmost derivation you want to do. So, this B we can replace by either one of this, but you have to have this one. So, we will take B is going to 0 S this move. So, 0 I sorry 1 S B now again this S we can go to we have all the one option 0 1 0

BB ok. So, 0 1 0 BB; so now, we can have. So, now, after this we can have this 0 1 0 BB and have a B over here sorry. So, now, we can take all the Bs to be 0's.

So, we can take first this B 0. So, 0 1 0, 0 BB, then you can take this B to be 0 then 0 1 0 0 0 B. So, this is the leftmost derivation then 0 1 0 0 0 0. So, this is a string in the language generated by this. So, this in short we can take say 0 1 0 to the power 4 ok. Now, we will construct the corresponding PDA which will accept the same language. So, how to do that?

(Refer Slide Time: 17:33)



So, we will just take M, q 0 1 is phi the way we constructed that is the only one state we have and this is the terminals those terminals are just the input alphabet 0 1 and then we have this stack symbol S B which are basically the v delta like this. So now, let us define the rules. So, rules will be coming from this; so now, S is going to 0 B this will give us the rule like this. So, we are a state is same, we have only one state now if we see a 0 and if the stack symbol is S.

So, this will be containing q BB this is q BB and we have from this we can say with the state is always same B is going to 0 S so that means, if we see a B over here, this will content that yeah. So, up with 0 we have this and this. So, here nothing means it is epsilon. So, it is q S for coming from this role by this production, and we have B is going to 0 from here we have q epsilon these two rules we have ok.

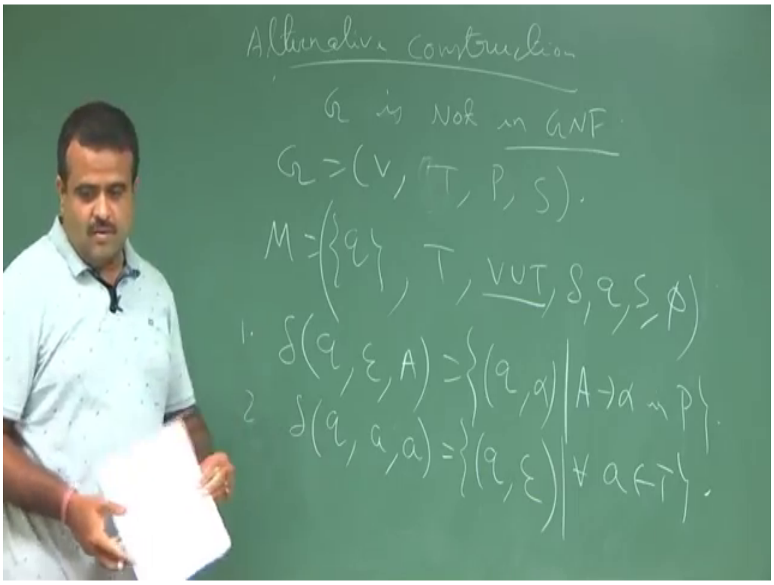
And, now we have a B is going to 1 S from there we can say that delta of q comma 1 comma B this is going to 1 S. So, that means q comma S. So, these are the rules. So, with this rule we will try to accept this string 0 1 0 to the power 4 we will try that with this transition rules we can accept this. So, for that we start with q we put this into that f and then our starting symbol is S ok. So, this will go to where? This will go to so 0. So, you first need a 0 q 0 s. So, q 0 S q 0 is equal to this so; that means, this S will replace by BB. So, q will be remain 1 0 to the power 4, S will be replaced by BB.

Now, we have a 0 over B here so; that means, delta of q 1 B. So, q 1 B we have a only one option so, q s. So, this B will replaced by S. So, this will go to q 0 to the power 4 and this B will be S B ok. Now again this will go like this we have a 0 over here. So, q 0 S so, q 0 is again it will go to q 0 to the power 3 now, S is going to replace by BB; so, three B's BB B. So, now again this 0 B so, 0 B yeah now 0 B we have few options, we have two options over here 0 B 0 B q 0 B we have two options, we can take S or we can erase that B so, we will do that. So, q 0 square B B and another way is q 0 square S B B two path is coming ok.

Now, let us follow this path, we need to converging on at least one path, then that will work. So, this will be like this. So, this is again q 0 B we can take this rule. So, this will be q 0 1 B this erased, then again we can take this rule q 0 B. So, it will be q epsilon epsilon; that means, this is accepted 0 1 0 to the power 4 belongs to N of M.

So, this is the construction for this. So, we can have another construction corresponding to this, but not in general I mean which may not be in the GNF. So, that is the all that is we are calling the alternative construction. So, this any string which are accepted by that grammar can be accepted by this automata ok.

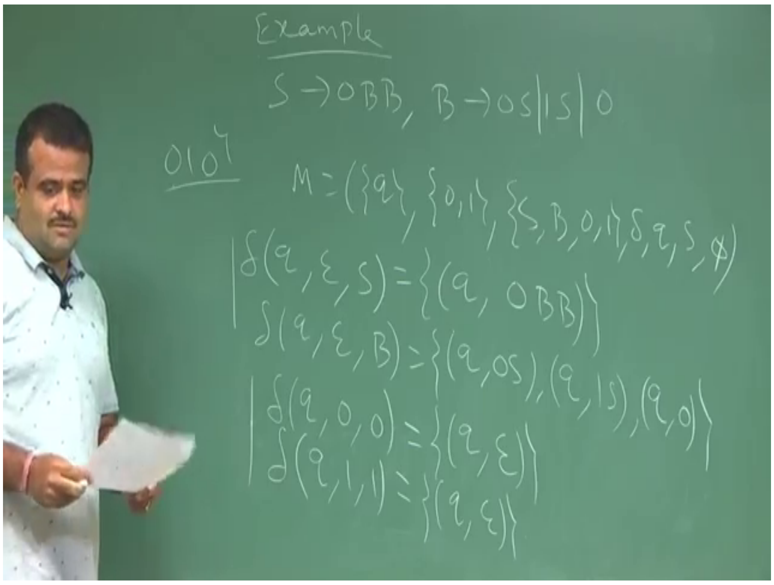
(Refer Slide Time: 23:27)



So, alternative j construction, alternative construction; so, suppose G is not in GNF or even if it is in GNF we will not consider that. G is not in GNF; that means, we have a given a G. So, this is T, now we construct a grammar a PDF like this. So, we will take a same, but only thing here the stack symbol we are going to put as V union T. Delta we are going to define q is the only state, S is the starting stack symbol like this and here we are going to define delta as like delta of q comma epsilon comma A is containing q comma alpha if we have a rule A is going to alpha in P.

So, here it is because our stack symbol is containing V as well as T. So, it is coming from V star I mean if V union T star and the second rule is delta of q a a this is nothing, but q comma epsilon if this for every a in T for all a in T. So, if you have a terminals then it will be q comma epsilon ok. So, with that we can have a construction like this, the same example we can take, but using this alternative construction.

(Refer Slide Time: 25:53)



So, S is going to 0 B B and B is going to 0 S or 1 S or 0 here we do not care about whether they are in GNF for not although, they are in GNF, but we will have this construction this alternating construction. So, we have a q over here and this is the 0 1 terminal strings and here the symbols are V union T. So, V union T is V is this and T is 0 1 that is the difference; delta S phi. Now, how we define delta?

So, delta of q epsilon S it is nothing, but q comma 0 B B, now we are allowed to use the string of terminals and the variables and delta of q epsilon B which is same as q 0 S, q just the rules just the production will give us this q comma 0 ok.

And for the terminals this is one and for the terminals what we said delta of q 0 0 is nothing, but q comma epsilon and delta of q 11 is nothing, but again q comma epsilon and we can easily check that this is accepting this string 0 1 0 4 that we can easily verify ok. So, this will be given in the lecture nodes.

Thank you.