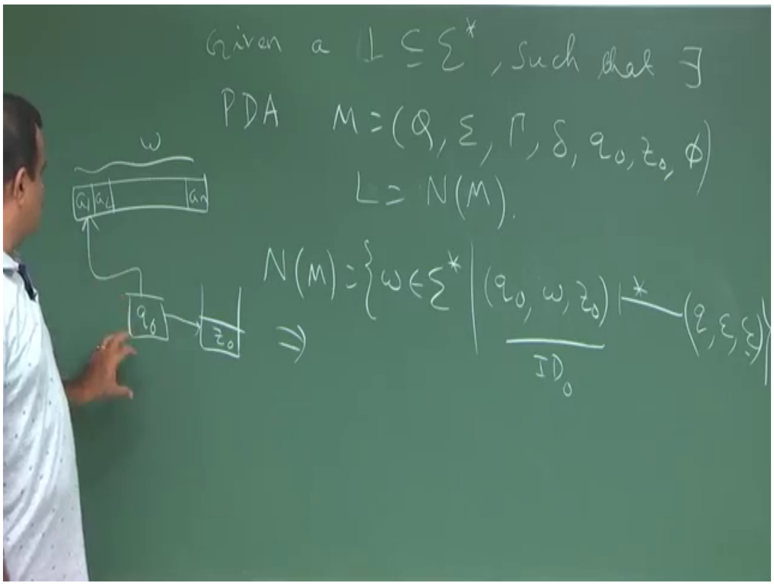
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 55
Equivalence PDA and CFL (Contd.)

So, we are talking about the equivalency between context free grammar and PDA. So, today we will talk about the given a PDA which is accepting a language. We can construct a corresponding context free grammar. So, then we will see the theorem which is still as their equivalency.

(Refer Slide Time: 00:37)

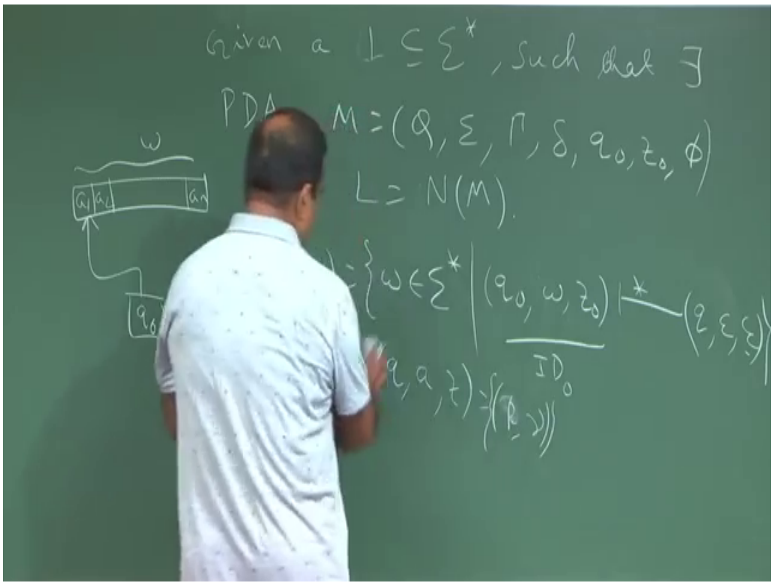


So, the theorem is telling given a given a language L which is a subset of this you know sigma is the set of input alphabet, such that there exists a PDA Push Down Automata in which is we know Q sigma delta q 0 z 0. We take the set of final state at phi, because we want to this is we are accepting in the term means of the empty stack.

Then such that there exists a PDA such that L is equal to sorry a N of M. So, N of M means set of all string, I mean this is the language accepted by this PDA in the sense of empty stack. That means, N of M you know N of M is the set of all string coming from sigma star, such that we start with q 0 and z 0 sorry this is the w, w is the t f.

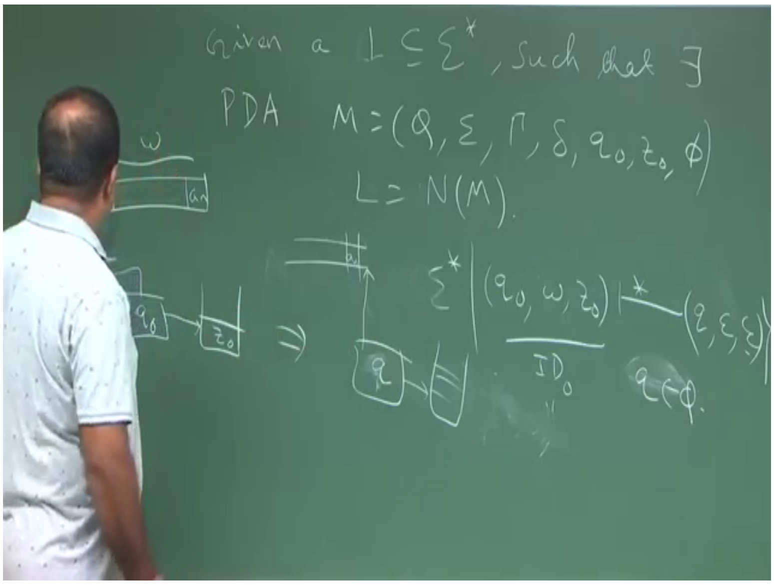
So, this is the initial ID, ID 0 so from this ID if we can go to the empty stack. So, there is no concept of final state over here we go to some state, then we completely reading the tape; that means, this w is exhaust and we reach to a empty stack so; that means, we put this, this is our tape this is a w a 1 a 2 a n say w is of length n. And this is our starting state q 0 this is z 0 this is the first, this is the situation of the initial snapshot initial ID.

(Refer Slide Time: 03:13)



Now if it is by we take the move up delta so then we go to some other state and it will keep on reading like this and the stack symbol will replace. So, we know the delta is basically a function of q we take a input alphabet and z between replace to, it is a non deterministic move sum p and sum gamma this way we have discussed.

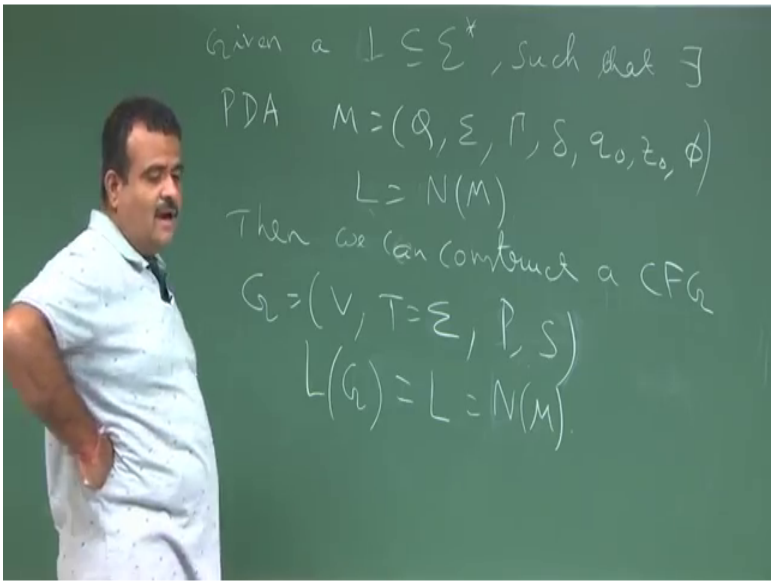
(Refer Slide Time: 03:32)



So, finally if it is reaching to a situation that, the stack is empty and the exhaust the reading this a n. So, our tape is pointing here and this is our some state q, this is for some state q for any q I mean q belongs to Q is any state we do not need to here you we have no concept of final state because this is the acceptance by the means of empty stack. So, this is done.

So, we have exhaust this; and now if the stack is also empty nothing is there. So, this is empty. So, then we called this w is accepted. So, such of all w is the language accepted by; that means, of by the means of empty stack. So, now, we will see we can construct a corresponding grammar, which will accept this language by the means of this CFL context free grammar.

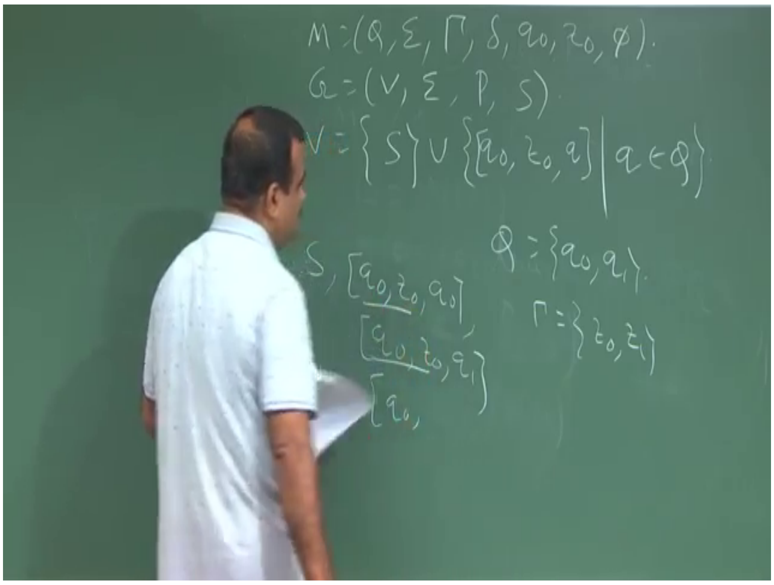
(Refer Slide Time: 04:46)



Context CFG we can have. So this is given, then one can construct then we can construct then there exist a we can construct a, context free grammar G which is we know V T, T is here if sigma because sigma is the terminals or the input alphabet ok.

And we have a starting symbol and we have P; P is the I mean rules we have a grammar sorry P is here P comma S, S is the starting symbol. So, S belongs to V such that the language generated by this grammar is same as L which is same as N of M ok. Now we will see that construction of such a grammar. So, from a given PDA how we can construct such a grammar that construction we will see. So, for that you need to define the symbols I mean variables, terminals is same that is the sigma, but variables we need to defined ok.

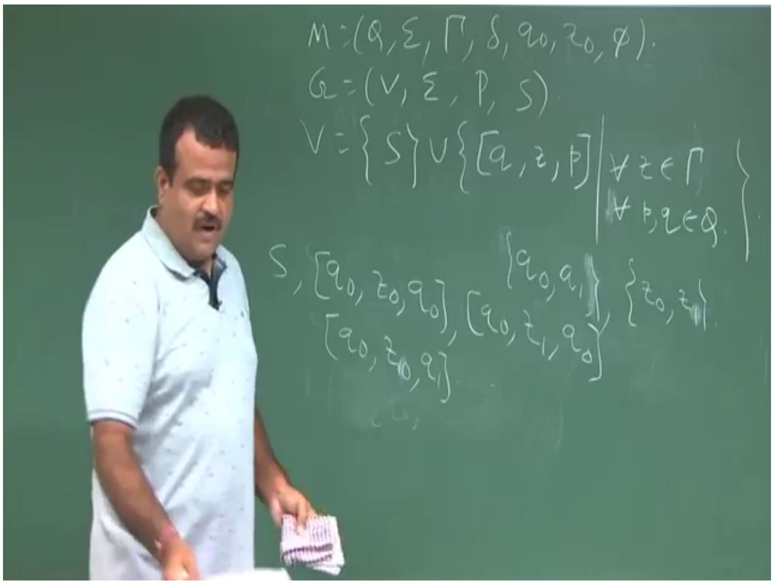
(Refer Slide Time: 06:32)



So, let us try to construct a grammar. So, we have given this Q sigma, this is the stack symbol we have delta q 0 z 0 phi. Now, we defined a G which is v then sigma is our terminal string I mean terminal symbols which is we are taking a sigma, then we have a P, we have S; S is the starting state.

Now, if V consist of S along with this type of tuple, 3 tuples like this q 0 q 0 is the starting state z 0 is the starting stack symbol and the q and this is for all q belongs to Q 0. So, if we have say two states q 0 q 1, then what are the states are I sorry what is the variable variables are say S comma all these symbols. So, z 0 if we have say z 0 z 1 then we have z 0 q 0 all possibilities q 0 z 0 q 1 q 0 this q 0 z 0 is fix over here sorry you know not z 0 fix. So, z 0 will vary. So, Q sorry now this is not we have to vary this is the movement. So, we have to vary all possible states and the stack symbol.

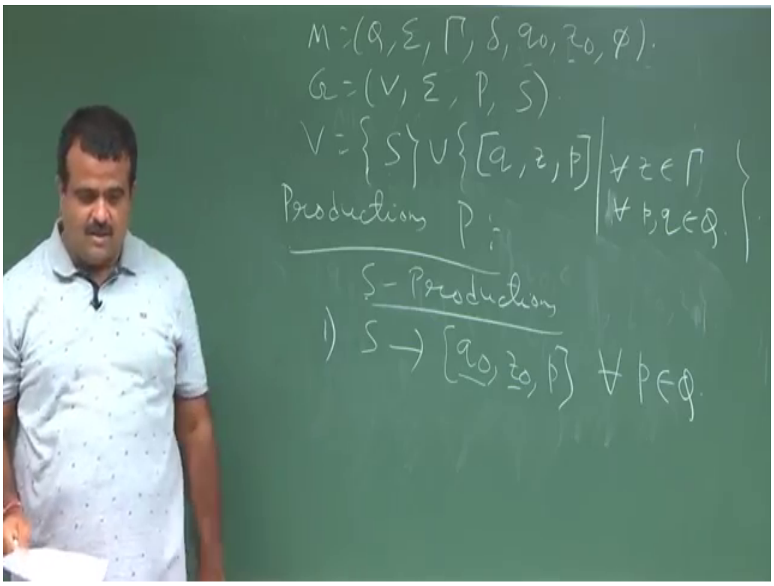
(Refer Slide Time: 08:34)



So, this is basically setup, I mean these three tuples q z p where for all z coming from the stack symbol and p q r for all p q coming from Q.

So, all possibilities are there. So, if there are two states q 0 q 1 and if there are say two symbol z 0 z 1. Then what is our V? V says along with this all possibilities q 0 z 0 q 0 q 0 z 1 q 0, then we have q 0 z 0 q 1. So, all the possible combinations of this coming from this so this is the way we defined this. So, now, how to construct the rules now? That you have to see.

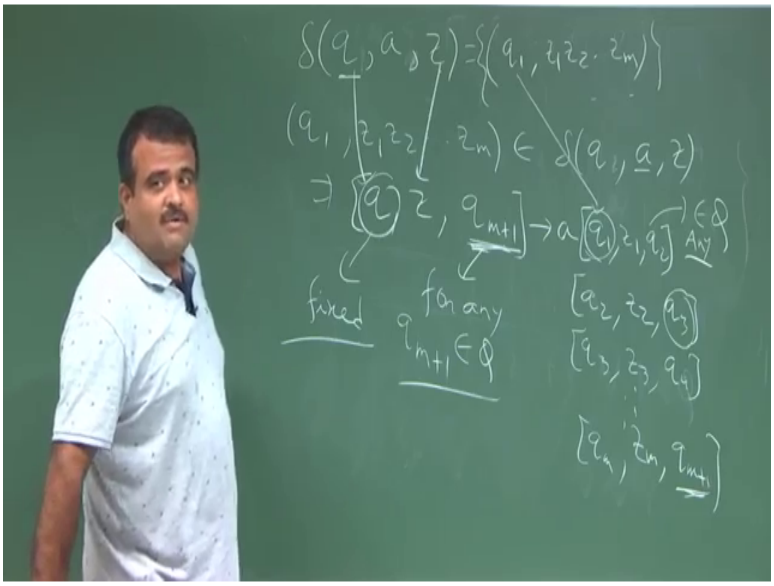
(Refer Slide Time: 09:55)



So, to construct the P we will take S first, S production this is for S productions. So, for S production we have this rules like this. So, S is going to q 0 z 0 p. This is for all p belongs to Q ok. This is for all p belongs to Q. So, this is if, so these two are fixed these two are coming from this q 0 z 0 and p we are varying. So, this is the way we construct the S production.

Now, we will see the other production, like we have that other production and this is the S production. Now we will see the other productions R S production will be coming from the transition functions.

(Refer Slide Time: 11:16)



So, transition functions means suppose we have a transition like this, delta of where are q and a is the input symbol z is this. So, suppose this is having say this is going to say some q 1 yeah q 1 z 1 z 2 z m. So, there are may be others also so; that means, this q 1 z 1 z 2 z m is belongs to delta of q comma delta of q comma a comma z. So, this will yield some productions in the P.

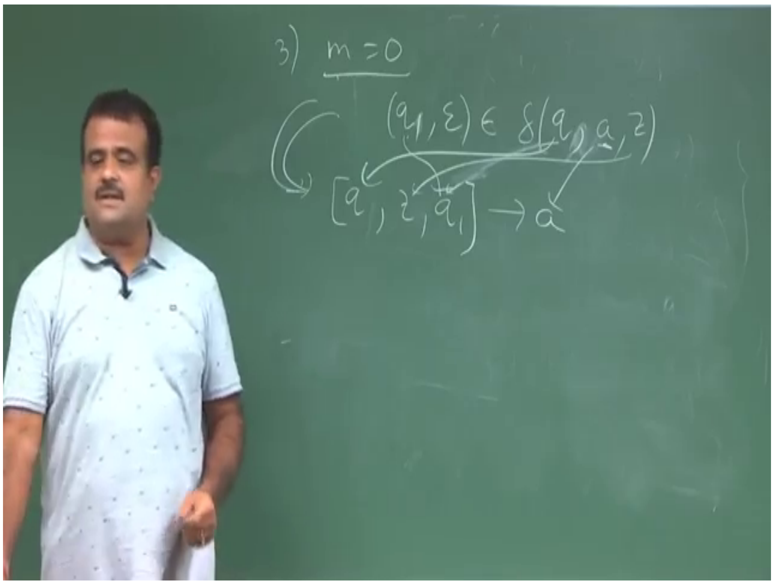
So, what are the production it will yield? It will yield the production like this. So, this q; q we are fixing this q and z here we are fixing this z this q is coming here this z here fixing.

And now, we have a choice of this q m plus 1, this is any state we can say, we have a choice of q m plus 1. So, this is going to a this a along a. So, we start with q 1 v 1 we have no choice this q 1 will come here, q 1 and then we have alternative z 1 then we can take q 2 we have a choice over here, followed by these are the product. Then q 2 we have to take z 2 then q 3 q 3 z 3 q 4 dot dot dot, then we have finally, q m these are the choice I mean these are the all options, these are any states this q 2 is any state, but if this is q 2 this has to be q 2.

Then q 3 of so any states within q m, z m and q m plus 1 this q m plus 1 and this q m plus 1 are the same like this, so dot dot dot these are the production we have. So, we have choice here we can choose this for any; so we have choice here we can choose these fixed this is fixed and this one is fixed these two are coming from these rules, now this is q m sorry this is q, this is a this is comma and this is q 1.

So, q 1 z 1 this and we have a choice over here. So, this we can choose this belongs to Q for any. So, we have a option there once this is fixed these are two be fixed then we can choose this is fixed, so this way we can continue. So, these are the rules for the productions these are the productions coming from this transition like this so this is clear. So, now, we have to have if this is having epsilon, I mean if this is erasing then what will be the move? So, that you have to if m m is equal to 0 m is equal to 0 means nothing is there. So, this is going to epsilon so that you will see.

(Refer Slide Time: 15:03)

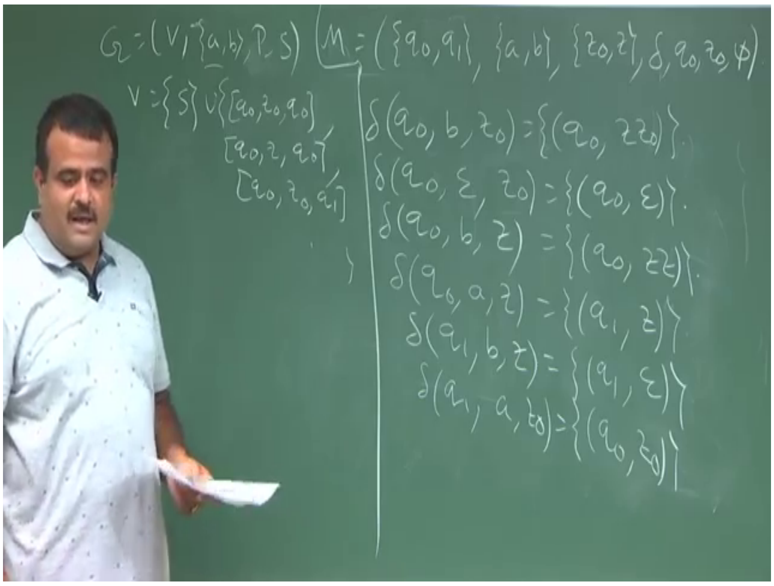


So, let us say this is rule 2. So, if m is equal to 0; that means, if q 1 epsilon is belongs to delta of q a comma, z comma z ok; that means, this z is erasing we are erasing the z, we have to erase the z, erase the symbol to get the empty stack so that we are doing so ok.

So, for that what is the rule? So, as we take the q over here and then, we have z and we take the q 1. So, suppose this is going to q 1 this is going to a. So, this will yield us a rule like this is going to a if it is erasing. So, q of z q 1 is going to a this a, q 1 is this one q sorry q 1 is this one, this is q this is q 1 and this is z this is our a ok.

So, this if we this is the construction, this is the construction of the grammar. So, if you construct the grammar in such a way then we can prove that, this will give us this will generate the language which is because we are emptying this by taking the terminals, terminals are coming from this input alphabet. So, let us take an example then it will be more clear.

(Refer Slide Time: 17:06)

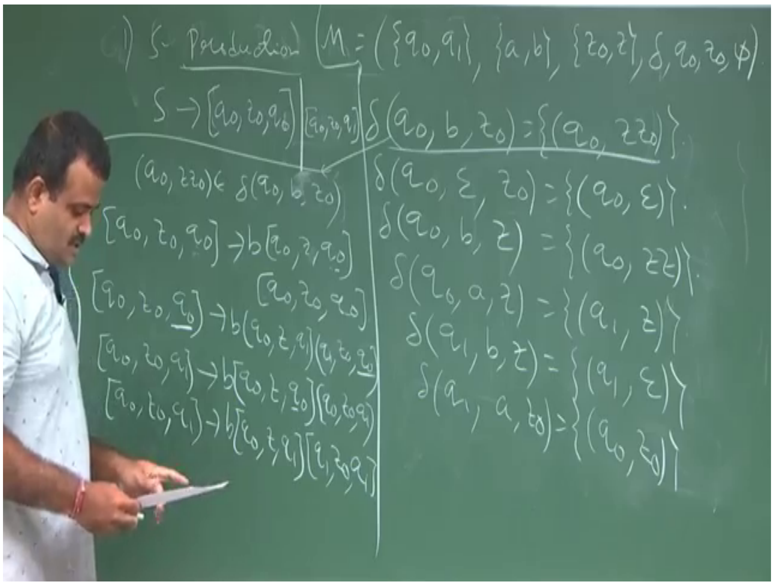


So, let us take an example so consider a PDA which is having two state q 0 q 1 and say a b, this is our sigma. And we have z 0 z two variable in the stack, delta we have to defined q 0 z 0 phi ok. This is the and delta is given by this delta of q 0, b, z 0 is q 0, z z 0. Then delta of q 0 epsilon that is epsilon move over here, z 0 this will give us q 0 epsilon. Then delta of q 0 b z this is nothing, but q 0 z z. Then delta of q 0 is it this is nothing, but q 1, we have z will be remain as z, but it will change to q 1. Now then let me complete this delta of q 1 once I have q 1 d z it will give us q 1 epsilon ok. It will erase this z like this ok.

So, delta of q 1 a z 0; which will be q 0 z 0 ok; so, this is the rules. So, now we to convert this to a grammar; so this is given PDA, now the corresponding grammar we know G is V T T is here a b and we have P and S.

So, basically we have to defined S, I mean we have we have to defined the V and S. So, as we said T is given this and a V is basically S along with all the tuple like this, p z q where p is p q r coming from the q set of all possible states and z is the input alphabet. So, if you do that then the states are basically I mean variables are basically this. So, q 0 z 0 q 0 then q 0 z q 0 all the possibilities then q 0 z 0 q 1 like this, all the possibilities I am not writing all over it is written in the lecture nodes, so it is there.

(Refer Slide Time: 20:58)



Now we have to define the rules the S rules at the others productions. So, rules will be coming from here. So, let us write the rules, so S rules as you know S rule is nothing, but so S production. So, S production as you know it is just S is going to q 0 z 0 p and p here it is vary. So, we have two states; so it is give us basically S is going to q 0 z 0 q 0 q 0 or S is going to q 0 z 0 q 1. This is one rule, this is two rules we have S is going to this S is going this is the S rule.

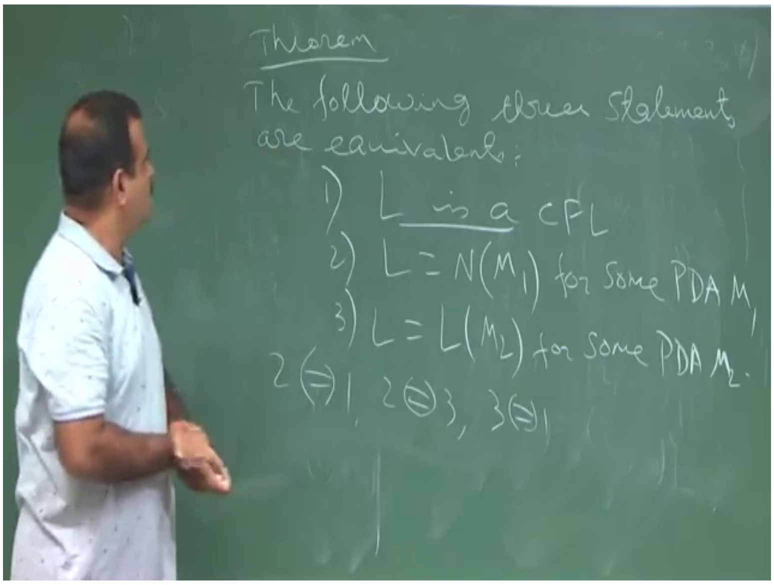
Now, the other rules are coming from the transition. So, let us take this transition. So, we will see how this transition we will covert. So that means this belongs to that q 0 z z 0 belongs to delta of q 0 b z 0. So, this will give us this rule, so q 0 z 0 q 0 will go to all possibilities, the b is here b of q 0 z q 0 along with q 0 z 0 q 0 ok.

So this we have a choice, here we do not have any choice, this we have a choice and this we have a choice. So, once we fix this so once we fix this will be like this, so this is one rule. So, another rule is so in vary this one. So, another rule is q 0 z 0 and this is q q 0. So, we can vary this also. So, we can make this q 1. So, is going to b of q 0 this is z 0 sorry, this is earlier one it was z because z is the first one z tan, q 1 then we have to take q 1 first then z 0 q 0, because whatever the over state here over here that will come here.

And another two possibilities is we can take these two as a q 1. So, this is q 0 z 0 q 1, this will go to b of q 0 z we can take here also you have the option we have take, we can take q 0, and then we can take them you have to take q 0, then z 0 and then q 1. And another option is here we can take q 1. So, that is the q 0 z 0 q 1 here also we can take q 1 b of q 0 z q 1, then q 1 z 0 q 1.

So, this is the rule corresponding to only these transactions ok. Similarly we have to have the rules for all these transactions, then we will get the corresponding grammars and we can see their equivalence they are accepting the same language. So, this is the details of this is given in the lecture not I am not writing all others so ok. So, they are basically equivalent, now we will have a theorem, which is telling the equivalency between all of this PDF and CFL ok.

(Refer Slide Time: 25:16)



So let us have this theorem, sorry the following three statements are equivalent, first one is L is a CFL, L is N of M 1 where M 1 is a PDA, for some PDA M 1, L is L of M 2 for some PDA M 2 ok. So, just now we have seen which one given a this one we can construct this.

So, just now we have seen 2 to 1, and we know if a language is accepted by a PDA in the means of empty stack, it can be accepted by a PDA and we can construct the corresponding PDA which will accept by the means of final state. And we know given a CFL we can construct this, so these are all equivalent basically, 2 3 are equivalent then 3 to 1 like this, so this we have already seen.

Thank you.