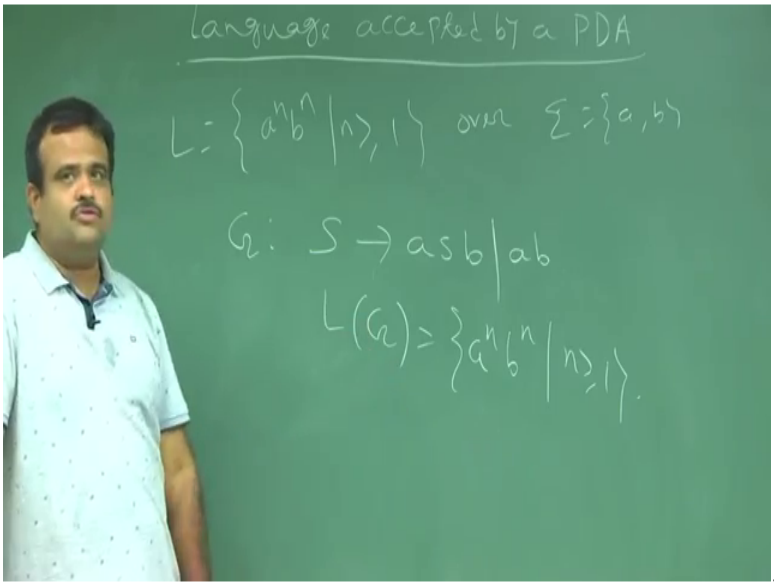
Introduction to Automata, Languages and Computation

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

Lecture – 49
Language Accepted by PDA

(Refer Slide Time: 00:15)

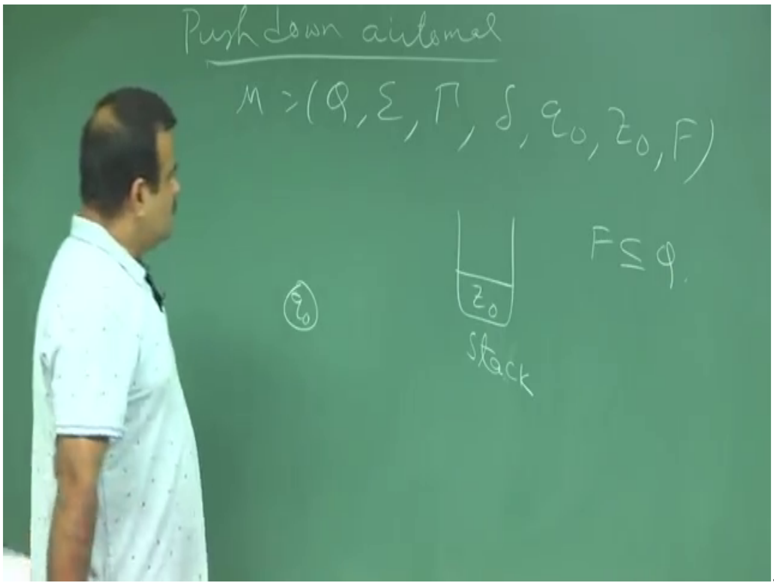


So we are talking about pushdown automata. This is basically to accept the class of language which is context free language. So, we have seen the regular language I mean we have seen some of the language which are not regular, that means, we do not have any finite automata which can accept this, we do not have any DFA or NFA or epsilon and if you like language like a to the power n, b to the power n while n is greater than 1. So, this type of language over sigma0 1.

We know this is not a regular language that means, we are not having any finite automata which can accept this language. So, but we do have a context free grammar to accept this language S is going to a s b or a b. So, the language of this grammar, if this is the grammar I mean this is the rules of the gummers, then we have seen the language of this is nothing but. So, we do have a grammar for this. Now, we will talk about the whether we can get to finite automata I mean this is called pushdown automata which can accept this.

To accept this language what we need to do we need to store the how many times you are encountering the in how many times this was a is coming, then followed by how many times b is coming this number should be same. So, for that we introduce the what is called pushdown automata, there we are using a stack extra stack. We can say that is to memorize or some sort of memory.

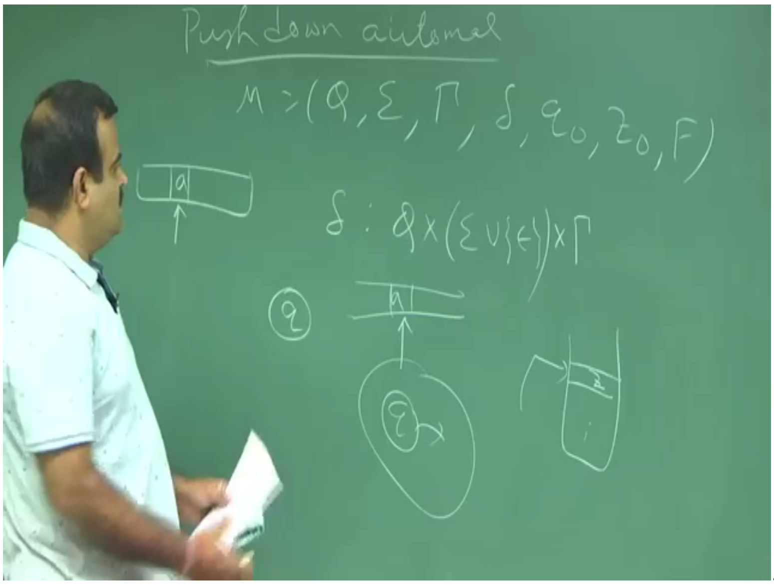
(Refer Slide Time: 02:11)



So, what is the pushdown automata? So, just to recap it is a tuple like this, Q, sigma, delta, q 0, z 0, F. We have seen this in the last class we define this formally. So, this is the set of states, this is the input symbol, all are finite. So, this is the stack symbol. Now, the stack is the new thing which is introduced than the waiver in the frame, this is the transition rule.

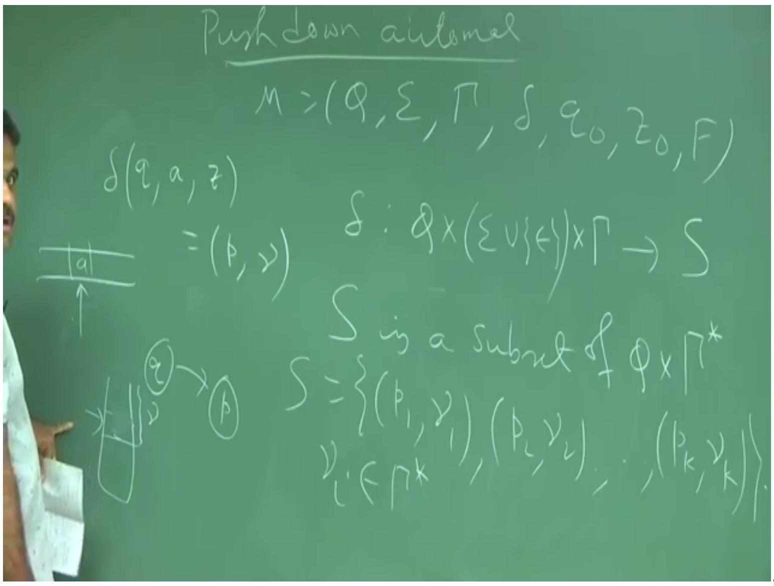
And this is q 0 is the starting state and z 0 is the starting symbol in the stack. So, every stack is initialized by z 0. And we have this q 0 is the starting state of this, starting state of this. And F is the final state; F is a subset of q ok. And then we have seen this say this is basically a non-deterministic form of this.

(Refer Slide Time: 03:27)



So, delta is a transition function which takes the we are at some state q. And we can read some input alphabet say this is a, we are reading some a. So, and it is a, this cross some alphabet in the stack top of the stack. Suppose, this is the position of the stack this is z, it is reading the stack z is the current symbol of the top of the stack and it is reading the tape like this, input tape. And we are at q 0, so it will move to.

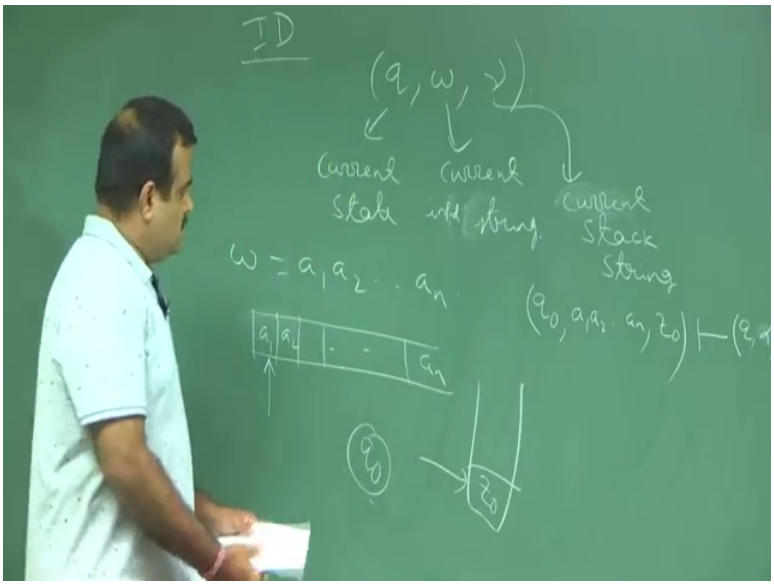
(Refer Slide Time: 04:23)



So, in a non-deterministic sense, it is a subset of; it is a subset of, this will go to it is a subset of this will go to some set S, we refer S. S is a subset of Q cross the string of stack symbol like S could be like this. So, p 1, gamma 1, p 2, gamma 2, p k, gamma k, where p i p 1, p 2, p k are the states, and gamma i is a string of stack symbol. So, that means, if it is going, it will go to any one of these, suppose it is going to suppose delta of q comma a comma z, suppose it is going to some e comma gamma.

That means, what that means, it is reading a string a, a is here in the tape. And we are at q state and our stack symbol is like this, this is the position of the stack; top of the stack is z, this z. And now it will go to some state p and this stack symbol is replace by gamma. So, gamma is a string of stack symbol and it will move to the next position of this. So, this is the, this we have seen ok. Then we have defined the instantaneous description of the automata that means, that is a snapshot of this automata.

(Refer Slide Time: 06:11)



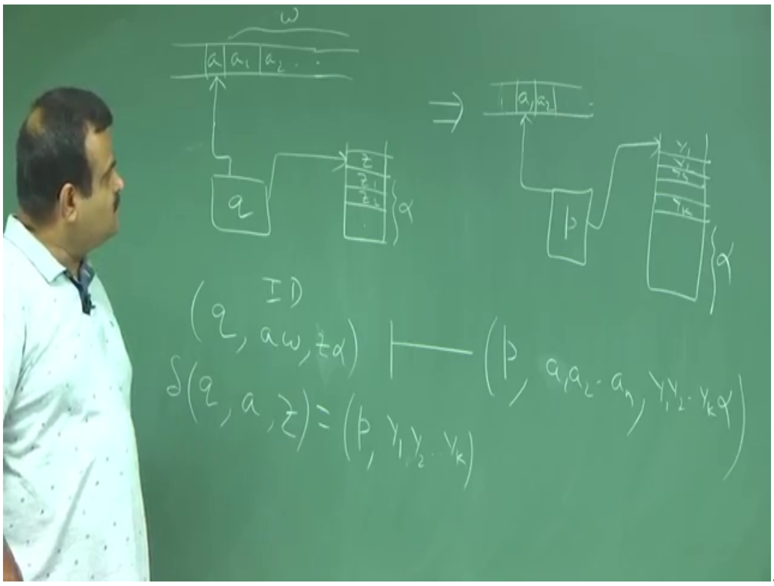
.

So, it is of this form like this. So, ID Instantaneous Description of your PDA. So, it is basically a tuple like this, q, w, gamma. So, w is the q is the current state, where our machine is this is the current position of the tape which is yet to read, current tape I mean input tape currently input tape. And this is the current position, current position of the stack string, input string; input string. And this is the current stack string ok.

So, suppose our input is we want to take an input like a 1, a 2, a n this is the input. And we are starting from q 0. So, our we put this in a tape a 1, a 2 like this a n and we are starting from q 0 and our stack is like this z 0 that is the initial symbol. So, this is that. So, this is pointing here and this is pointing here this control ok. So, this is the tape position which is initialized by the this thing and initial state is this and z 0 is the initial stack position. So, from this, we are going to move. So, this is the snapshot of the current position of this, this is the initial ID.

So, initial ID is nothing but q 0, a 1, a 2, a n this is the tape position and z 0 ok. Now, this will go to this is the move relation, this will go to some state in this form q some other w or some alpha, then some gamma 1 like this ok, so that we have to that is called move relation in the ID. So, this ID is going to next ID.

(Refer Slide Time: 08:59)



Like if we have if our control is like this, we are reading a tape like this. So, we have a, a 1, a 2 and this is the current tape header. And we are at state q and this is the stack position, we have some symbols say top is z, this is z 1, z 2 dot dot dot ok. And suppose it is so this is the control unit this is pointing this a and it is pointing the top of the head. So, this is the current ID.

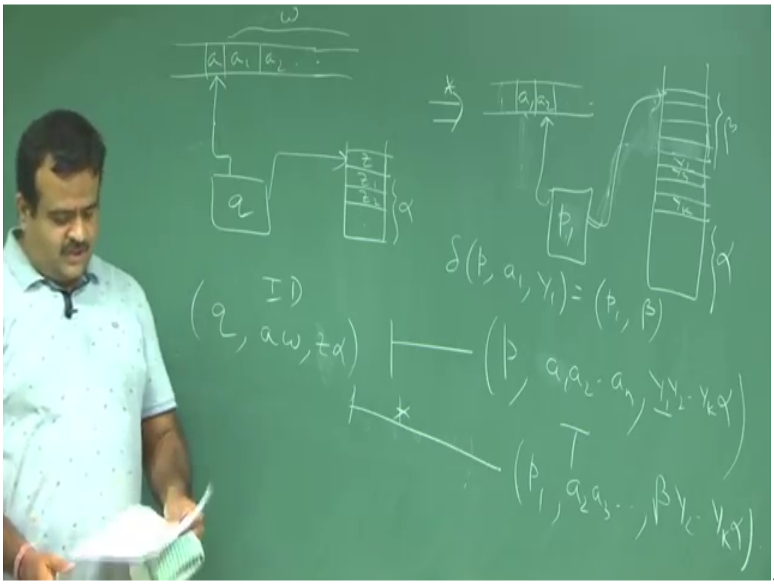
So, current ID is q a and if you say this is w, a w, a is input alphabet and the current position of the tape stack current position means the total stack. So, that is basically z is the top. And if we denote this by alpha z alpha, this whole string if we denote by alpha.

So, now if this is going to in the delta rule, if the delta is like this. So, delta of say q a this reading a and it is saying z, suppose this is going to some p, it, it may have many options, because this is non-deterministic, but we are taking one option suppose it is going to p, and then the stack position will be say y. So, z will be replaced by say y 1, y 2, y k ok, so that means, this will go to I mean the next move, next move next ID will be like this. So, this step is pointing now, this is a 1, because this a is gone, a 2 like this and currently we are at p and our stack position is like this.

So, up to this we have alpha and this z is replaced by y. So, this is y 1, y 2, y 3 dot dot dot y k and this is pointing the top of the stack. So, this is that snapshot of the current position after doing this transition. After doing this transition our system our automata PDA moved to this situation. And we take a snapshot this is the current ID, so that means, we write relationship move.

So, this is going to the state is changing to q to p and the current tape is w, a we have already did. So, next so that is w or a 1, a 2 dot dot dot and the current position of the stack. So, current position of the stack is y 1, y 2, y k along with this string alpha. So, this is the move this is the move relation, we are at this ID to that ID, it is basically a snapshot of the we are taking a snapshot picture of the current position of this ok.

(Refer Slide Time: 12:43)

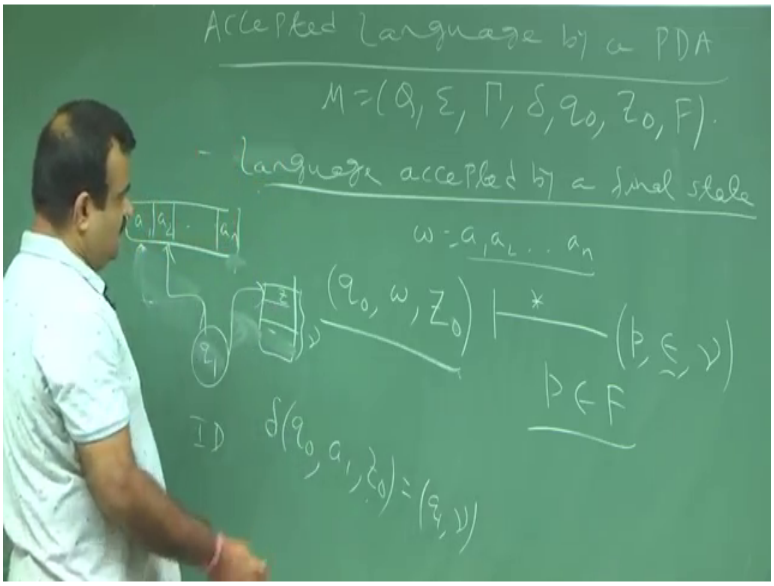

.

And next if we go from here, so now, suppose we are reading, now we are reading what, now we are reading this one. Suppose delta of p and you are reading a 1, in the input tape and we are reading y 1, suppose this is going to say some symbol like p q. So, say this is going to p 1 and the tape position is say, so this will p 1, so if the delta rule, so p 1. And y 1 is suppose replaced by some beta ok, then we say this is going to p 1 and then we have already read a 1 so a 2 a 3 so on and the stacked position is instead of y 1 we have beta, beta is another string of stack.

So, stack symbol beta y 2, y k alpha ok. So, this is the from here to here, it is to move and this is basically. So, we have here this is now pointing this and this is now p 1. And instead of this y 1 we have some other symbol of stack, this is beta this is beta, and this will be pointing this. So, this is the current ID, current position of this. And the from here to here, we are having only two move.

So, this is from here to here we can have more than one move relation we can go is this ok. So, this is basically that ID, I mean we can we are at current position of this system, then based on the input alphabet and the stack position we are slowly moving to the next ID I mean next position, next snapshot of this automata push down automata ok. And this is from here to here we need to have two relations. So, this is a reflexive closer like if we move from here to here, then there to there like this. So, on this using these will define what do you mean by the acceptance of a accepted language by a PDA with the help of this ID.

(Refer Slide Time: 15:13)



So, accepted language by a PDA. So, PDA what PDA except that is the point and yeah. So, this is basically again a 5 tuple given a PDA sorry not 5 tuple it is a 7, 7 tuple. This is input alphabet, this is stack symbol, delta, q 0, z 0, F ok. Now, we define this by two ways of language accepted by the PDA; one is accepted by final state and second one is accepted by empty stack ok, so that is first we defined that accepted by the final state.

So, language accepted by a final state. So, this is similar to the DFA or NFA, so that means, we when we called a string is accepted by a final given a string w. So, w is a 1, a 2, a n. So, when we say this w is a accepted string by the automata, if we start from q 0 if we read it and if we finally reach to a final state, any one of the final state after if there is one path which is reaching to the final state, after ending of this string reading of this string, then we call this string is accepted. And the collection of such string is called language of that automata.

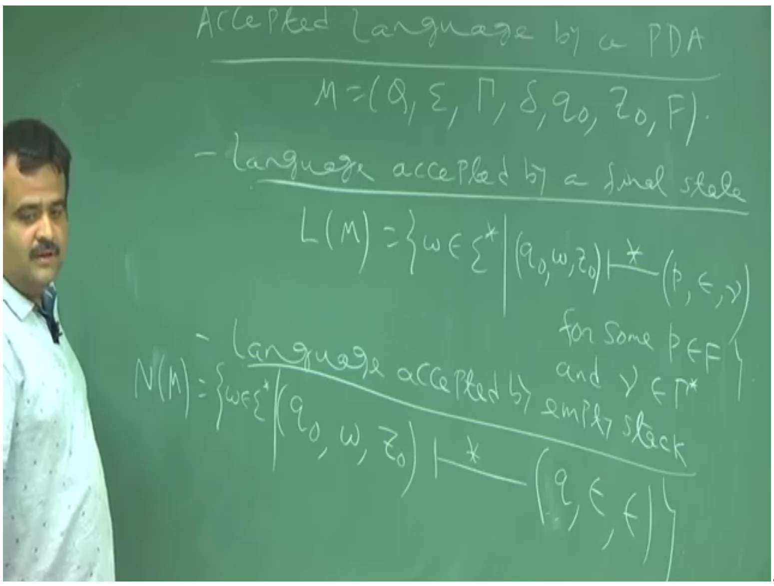
So, here also we defined like this. So, here we have to define in terms of instantaneous description. So, this string will be accepted by this PDA in terms of final state if we say we are starting with q 0 and we take this string as a input in a tape and our start symbol in the step is z 0.

If this is going to going to means one or more or many this relation move we have seen, this is a initial ID and it can go to some ID like this p, then string must be over input string must be over and we do not care about the stack, where p is belongs to one of the final stage. That means, what that is we are currently at p 0 and we have the this is the situation of the this is the current position of the tape a 1, a 2, a n ok and this is the stack symbol we are at z 0.

So, this is pointing here, this is pointing here. So, this is the initial ID, this is this one initial ID. Now, next we are at now depending on the delta of q 0 and a 1 z 0, we will move to someone, we will move to some p, some q 1 and this z 0 will be going to some gamma like this. So, this will be replaced by something some gamma and this is z. So, this will be q 1 and this will be pointing to next input alphabet, this will pointing the top of the stack. So, this is the next ID that means, it is first ID initial ID is going to this ID.

So, like that if you continue and if we keep on applying the transition rule, we will get the different, different ID, we are having that relationship move by this. So, eventually if we can reach to after ending of the tape, if the tape is over this is epsilon. Eventually if we can reach to a final state by the state transition functions, then this is called this string will be called accepted by this PDA in terms of accepted by final state.

(Refer Slide Time: 20:05)



.

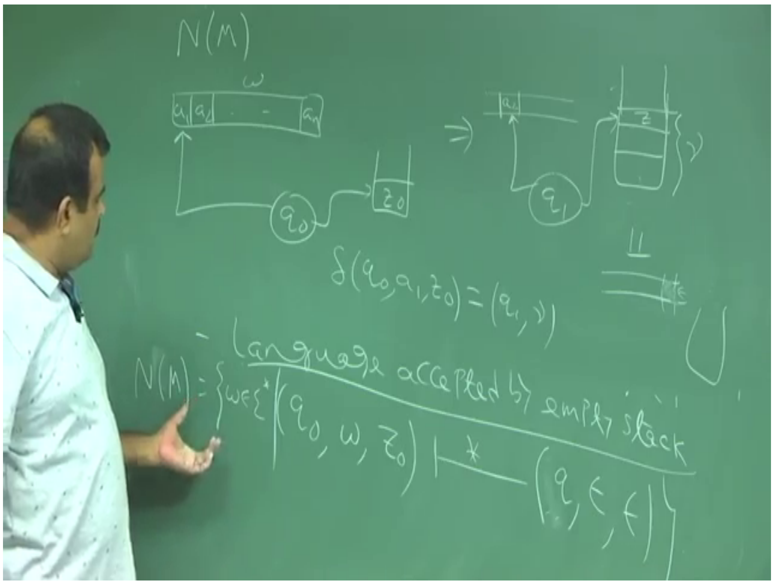
So, that is the idea. So, what is the language? Language we can define like this. This is a L of M, we can say language accepted by the final state. So, this is the set of all w string such that q 0 w, z 0 is going to p, some p, p is a final state. And this would be epsilon, that means, we experience since reading the tape and we do not care about the stack symbol, we do not want stack symbol to be empty in this case. So, for some for some p which is in the final state and some gamma in this ok, so, this is called language accepted by the final state.

Now, when we say the language, so this is one type of language when you say the language accepted by the empty stack, accepted by empty stack ok. So, this is basically we are starting from q 0 with the w and z 0. Now, this will be accepted by the empty stack if the stack become empty, and we do not care about the state where it is reaching, so that means, if it is going to some p, p may not be a need not be a final state here what we referred as q, q some state.

Input alphabet is empty we have already and the stack is also empty. If this is the situation, then I mean stack is empty that is called accepted by the empty stack. And we have already input alphabet this and q is any state, q need not be a final I mean f. So, this is we referred as N of M, this is set of all w such that this is happening.

So, set of all w such that if you start from q 0 with the input w, z 0 is the initial state in initial symbol in the stack. And if we reach to the if we reach to the empty stack after ending with the input alphabet, then this, this all the string of this type is called the string accepted by this DFA accepted by this PDA by empty stack ok. So, this is we refer as N of M. So, like this .

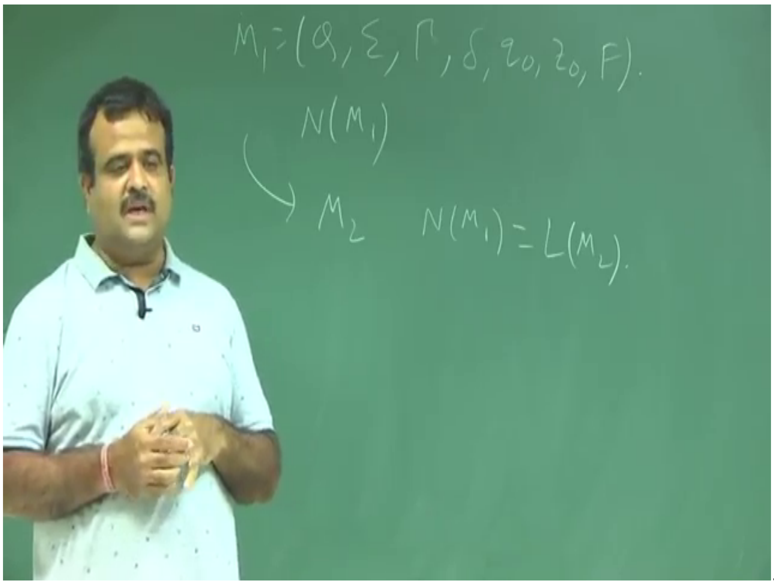
(Refer Slide Time: 23:35)



So, this is basically we start with N of m is nothing but set of all can we start with. So, this is our w a 1, a 2, a n. We want to accept this w by the empty stack. So, we are at q 0, and it is pointing here and our stack condition is like this z 0 this pointing here. And the based on the delta q 0, a 1, z 0, it will move somewhere maybe it will go to q 1 some gamma, so that means, this will go to. So, this will be the position now a two it will be pointing and like this will be q 1, because next state this will be point here I mean we may have other options also because this is a non-deterministic.

Then we can have here instead of this stack z 0 is replaced by gamma. So, gamma is say here. So, gamma is say some z like this, so, this is pointing here. So, like this if we keep on moving and until at the end of this string, when this stack is empty, so after this so it is after n it is epsilon only nothing is there, if it is reached to an empty string empty stack then it is called. So, eventually if it is reached to a empty stack, then it is called the language accepted by this PDA ok. So, later on we will see these two are same, that means, they are equivalent, that means, if a language is accepted by a PDA with respect to like.

(Refer Slide Time: 25:37)



.

So, given a PDA Q, then we have q delta, q 0, z 0, F. Now, given a PDA if it is accepting by a language, then from here we can construct n 2 such that another PDA similar to this we may use the same symbol only thing delta will be change we will see that. Such that L of M 1 is basically L of M 1 is basically L of M 2 and vice versa.

So, given a M, which is corresponding to a language accepted by means of the final state we can construct a PDF, PDA from here which will be corresponding to the accepted by the empty stack. And also we will see that these will be giving us the context free grammar. So, the PDA the language accepting by the PDA is nothing but it is a context free language. So, there is a context free grammar which in corresponding to that language.

Thank you.