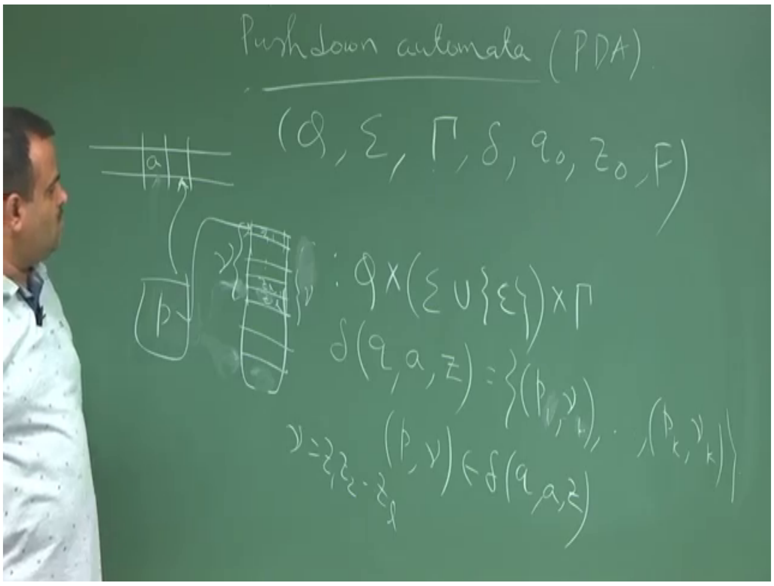
Introduction to Automata, Languages and Computation

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

Lecture – 51
Deterministic PDA

So we are talking about Pushdown Automata.

(Refer Slide Time: 00:19)



Just to recap. So, it is a couple like this, Q, sigma, gamma, delta, q 0, z 0 and F. So, Q is the set of states and this is the sigma is the set of input alphabet and this is the symbol in the stack. So, here we have a stack as you know and delta is the transition rules which is taking the input as a. So, the delta is a function form Q, we are at some state and we take a input alphabet in the tape along with the epsilon and then we read a top of the stack.

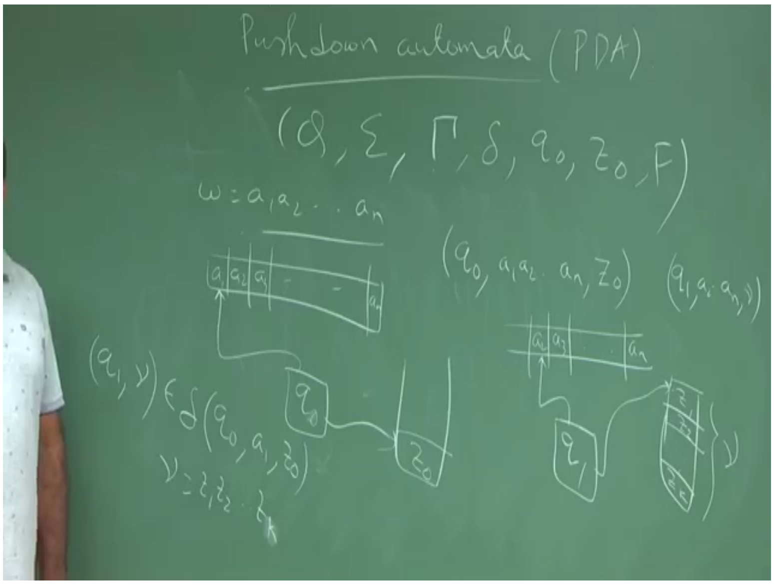
So, we have given a so suppose this is the input symbol in the tape and this is our finite state machine and we are at some state q and it is tape is heading here and this is the stack and which is initialized by z 0 and the initial state is q 0 and this is the initialized by z 0, this is the situation initially. And, initially nothing is there in the input suppose a is the input so, this is the initial situation. But, after that we have some stack symbol like z is the top of the stack and we have our state is at some time q; q is the state and this is the situation. So, this is the function of q, a state and a input alphabet a and a stack symbol z. So, stack symbol is coming from here and z 0 is the initial position.

So, these will consist of this is non deterministic. So, this will consist of some of the sets like p i, gamma, p 1 gamma 1 like this p k, gamma k. Now, suppose we have option here non deterministic move. Suppose p comma gamma is belongs to this and our PDA this is pushdown automata PDA our PDA is going to this state so; that means, this will be.

So, after this the new state will be so, a tape head will move to the. So, new state will be p and tape head will move here and this will be z will be replaced by gamma. So, say this is gamma. So, gamma consists of some z 1, z 2, z l say. So, this is a z l z del minus 1 dot dot dot say z 1. So, this is our gamma and then after that it will point to the again the top of the stack.

So, this is the situation, this is the movement by the PDA. So, it will move to the, it will replace the z the top of the stack symbol by string up stack symbol. So, that string is the gamma, it could be epsilon also. So, this gamma could be epsilon also. So, in that case that stack is stack symbol is erase kind of thing and it will move to the next part of this and using this we have defined what is the ID instantaneous description of a automata. So, that is the snapshot of the present state of the automata like we have the initially our initial state is yeah. So, we will define that. So, this is we have already discussed, but just to recap what is the idea again.

(Refer Slide Time: 04:41)



So, suppose initially we want to read a this is the our tape you want to read a 1, a 2, a 3, a n. Suppose, there are n input and we start with our journey from the state q 0 and the stack is like this. So, stack initial stack symbol is z 0. Now, it is pointing to this and the this.

So, this is the just if you take a picture snap sort of the automata this is the situation. So, what is the situation we have we are at q 0. And what is the tape? Tape is nothing, but this one w a 1, a 2, a n which you are going to read. And what is the stack condition? Stack is only one symbol which is z 0 this is the situation, ok. So, this is called instantaneous description. This is the initial ID if this is our w.

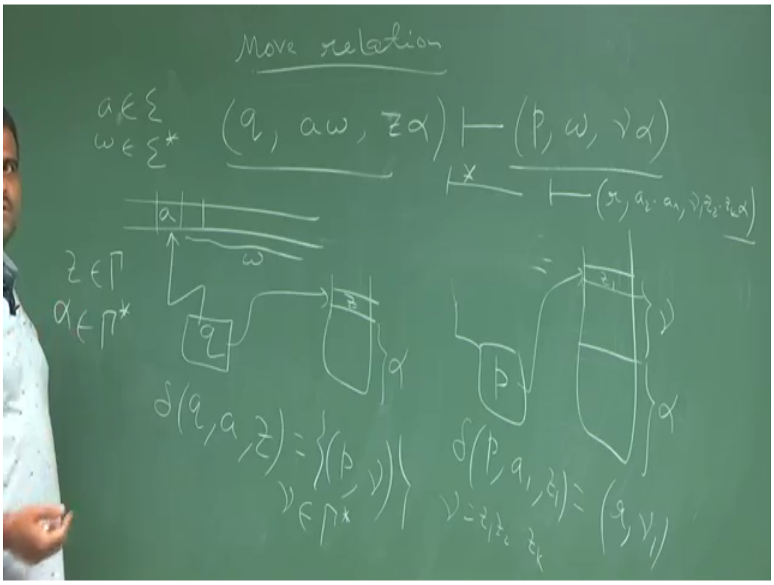
Now, suppose after this if we read delta. So, this is delta of depending on the so, q 0, a 1, z 0. So, depending on that if you will have a move. So, if suppose this is going to some of the state, suppose it is going to say q 1 and this will replace to this belongs to this and we have there could be many other option, but you have decided to go q 1 and this z 0 will replace by gamma.

So, what will be the next situation? Next situation like this; so, our tape we have already read a 1. So, now, our tape condition is like this a 2, a 3, a n and where we are? You are at q 1; q 1 is the next state I mean we were choosing that we have option and what is the stack situation? So, stack is gamma. So, gamma say consists of sum say z 0, z 2, say z l, or say z k. So, gamma will be replaced by z 1 gamma is sorry z 0 z k. So, this whole thing is our gamma and this is the top of the stack. So, it will point to here and this will point to here.

So, if you take the picture of this, now the situation is now our stack condition the automatic condition is we are at q 1 and our this is our tape and our yeah and our this thing is gamma. This is now current snapshot of the automata and we can say this ID; from this ID we are moving to this ID. So, this is the move relation from one ID to another ID. So, this we have already defined.

So, using this move relation we can define the language accepted by the automata and then today we will define; today we will show their equivalency and we will define the deterministic pushdown automata.

(Refer Slide Time: 08:18)



So, just to define formally the move relation; suppose this is our ID q, aw and say some z alpha. So, that means, this is the current situation; that means, what? That means, this is our tape, we have a and we have w over here the string and a. So, a is a input alphabet and w is a string of input alphabet; w could be empty also and z is a string. So, we have z is a sorry, z is not a string z is a symbol top of the stack and this is our alpha stack symbol and we are at our automata at q 0. So, this is the current ID current snap sort of the PDA, ok.

Now, suppose this p of so, we are now we need to take this delta move on this a. So, delta of q, a, z; so, it may have many options this is non deterministic. Now, suppose this belongs to it is having say, this option p it is q is going to p and this z is replaced by gamma. So, gamma is again a string of alphabet. So, here z is a alpha stack symbol I sorry gamma is a string of stack symbol and alpha is a string of stack symbol stack symbol, ok. So, this is the situation.

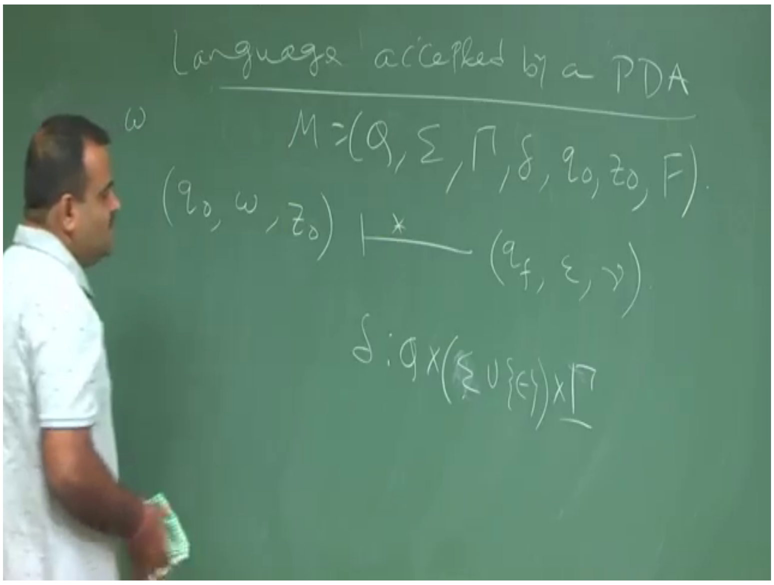
Now, suppose it is going to so this is going to a new state p and this we have already read a. So, now, situation will be w and then this z will be replaced by gamma. So, now, this will be the new snapshot. So, this is our w and this will be pointing here and we are at p now and our stack is like this. So, this up to this is alpha and this is replaced by gamma and this is the top of the stack and this is the situation.

Now, what is the current in a description current snap sort of this? So, our state is p and our tape is we have to yet to read w and the stack condition is the z was replaced by gamma. So, gamma is again a string of stack symbol; so, gamma and alpha. So, we can say this is the move relation. So, this ID to this ID we are moving by the transition rules of the PDA, ok. So, using this move so, we will keep on moving this.

So, again from this ID; if we move to some other ID like we take this delta p then say the opposite is a 1 and then suppose this is z 1 z 1 this may go to some other things so, that will be coming into the picture. So, that will be the move relation. So, if it is going to say some r and then say z 1 is going to replace by gamma 1, then we say this is going to r and say the suppose a 2 a 3. So, a 2 a 3 a n is the situation and the gamma is they still there. So, gamma is consist of say z 1, z 2, z k.

So, it will be like z 1 is replaced by again gamma 1. So, gamma 1 z 2 up to z k is there an alpha is filled again here, and we say from here to here we are moving in two step. So, we can say from here to here with me with more than one step so like this. So, this is the move relation recursively reflexive transitive closure, ok. So, using this move relation we have defined the language accepted by this PDA. So, there are two types of acceptance of a language.

(Refer Slide Time: 13:00)

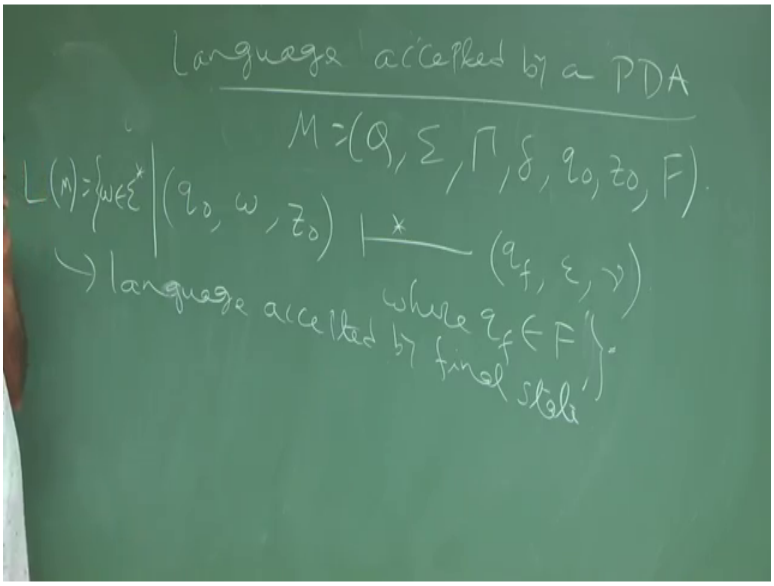


So, this just a recap this we have already defined in the last class in the last few lectures; language accepted by a PDA. So, we have given a PDA say Q, sigma, q 0, z 0, F; F is the final state, ok.

Now, our initial ID is we are at q 0 and we say w is the string we need to read. So, we put the w in the tape and the initial stack symbol we are using z 0. Now, suppose this is going to repetitive applying of this delta rules. If we can reach to a situation like this some q f and we must end up with the reading the string this is epsilon, this is epsilon and we have some gamma in the stack.

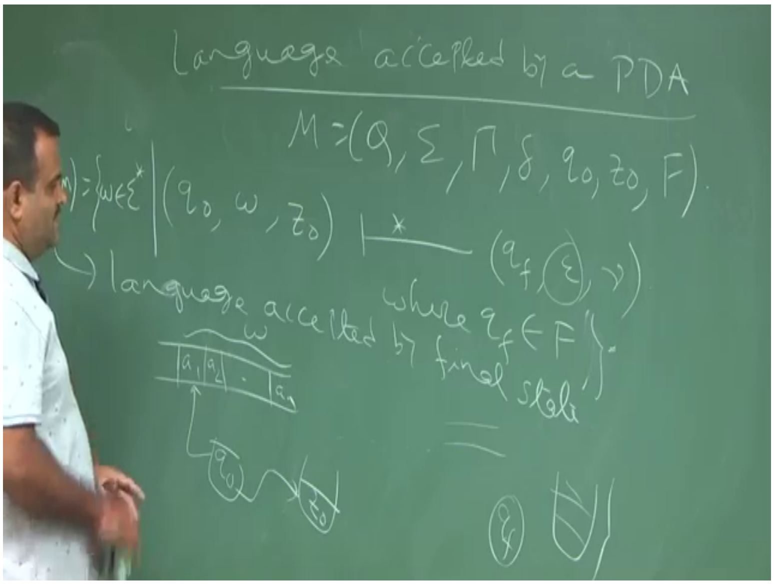
Stack should not be empty; once the stack is empty we have to hold there. There is no further movement. If the stack is empty because this delta is a function of Q then c input symbol including the epsilon and the stack symbol. So, once this is empty then we do not have any move because here we have no epsilon is there. So, then we have to hold there stop, ok, that is a stopping condition or halting condition ok.

(Refer Slide Time: 15:02)



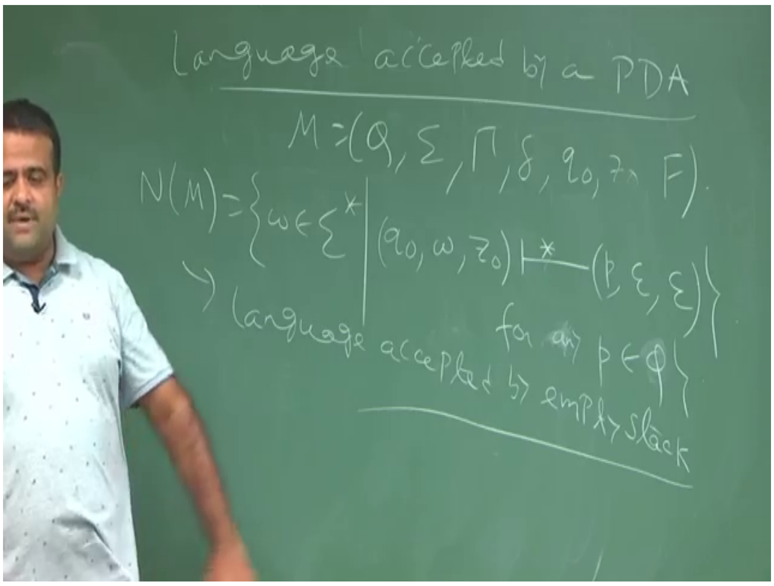
So, situation is like this. So, this is the, if q f is the final state then this kind of string is called language accepted by. So, this is set of all w sigma star such that this where q f is a final state, then we say this is called language accepted by final state, language accepted by final state.

(Refer Slide Time: 15:40)



So, that means, we start with this situation we are taking a 1, a 2 this is our say w and initially we are at q 0 and our initial tape condition is like this a stack condition is like this and we have keep on applying this rules and we are changing the IDs. So, the snapshot and slowly if we move to the situation, we exhaust reading the string there is nothing left in the ID in that this and if we reach to some of the final state and we do not bother about the stack because this is the language accepted by the final state. So, now we define the language accepted by the empty stack then the, so, we must emptify the stack.

(Refer Slide Time: 16:31)

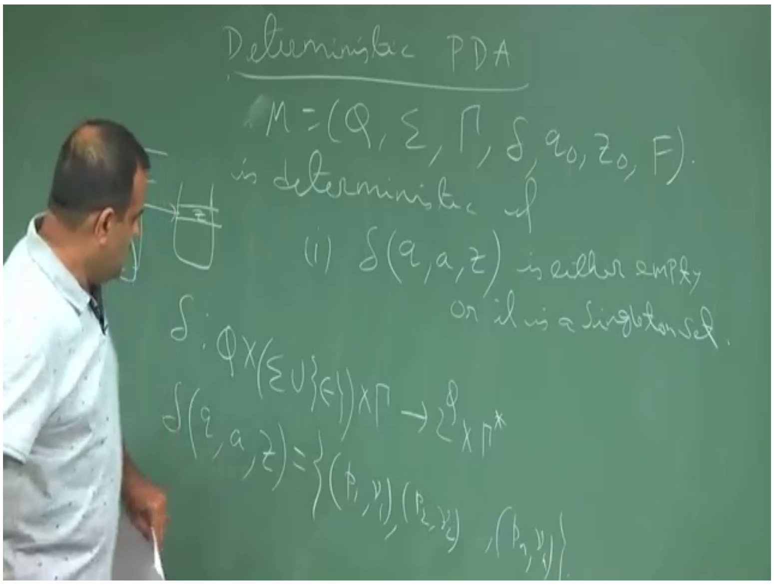


So, that is defined as N of M this is the set of all string of input such that start with q 0 with this w and z 0 we must go through we do not care about the state because this is not in terms of the language accepted by the final state. It is in terms of the language accepted by the empty stack. So, this is so, epsilon and this is also epsilon for any state for any state we do not care about this sorry q.

So, that means stack should be empty, I mean along with the once we finish the reading this what is called input tape input in the tape, then the stack should be empty we should be able to. So, then this is called language this is the language accepted by the PDA accepted by empty stack and this we will use in a more often because this will be we will show that the PDA is eventually accepting the context free language, CFL. There is a equivalency every for a given context by language you have a PDA which is accepting that language, accepting in the terms of empty stack.

But, we also show we also discuss the equivalency between these two; that means, if given a PDA which is accepted by the final state then we have a language which is we can have a PDA which will be accepted by the empty stack. So, this equivalency we will discuss, ok. So, this is just a recap. Now, today we will define deterministic PDA before we move to the equivalence here is of these two.

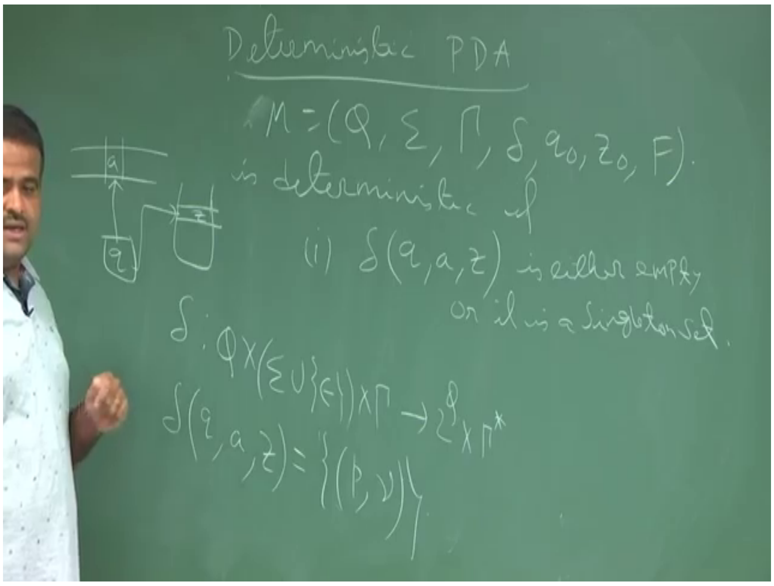
(Refer Slide Time: 19:05)



So, deterministic pushdown automata: so, a automata a PDA which is again this tuple delta, q 0, z 0, F is deterministic if I mean for non-deterministic the delta was a. So, what does delta for note in the PDA? So, delta is a function from Q cross sigma we can take epsilon also and this gamma and this is going to the power set 2 to the power Q along with gamma star.

So, that means, if we take a earlier one if you take a q and a; so, suppose this is the situation we have a and q is the current state and say top of the stack is z. So, we are reading z. So, q, z; so, for non-deterministic here we have many options like we have p 1, gamma 1, p 2, gamma 2, p n, gamma n ok. But, here for deterministic we do not have this, we have a deterministic move either this will be empty either this delta of q, a, z is either empty or it is a singleton set singleton set. Like this will be only one we do not have it is a deterministic move either we do not have any move or we have a deterministic move; that means, this will be like this.

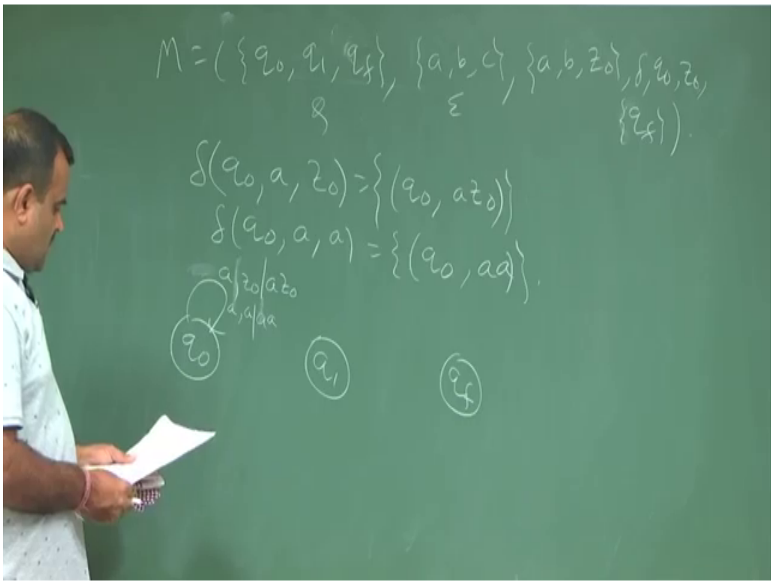
(Refer Slide Time: 21:35)



Say p and this z could be replaced by all the singleton set or it is empty. Then this is called yeah then this is called deterministic PDA; that means, this move is deterministic it has only one element in this thing.

So, now what will show; so, we can have some example of a deterministic PDA. Let me check if I have one. For deterministic PDA yeah, I think I have one let me take this example.

(Refer Slide Time: 22:35)

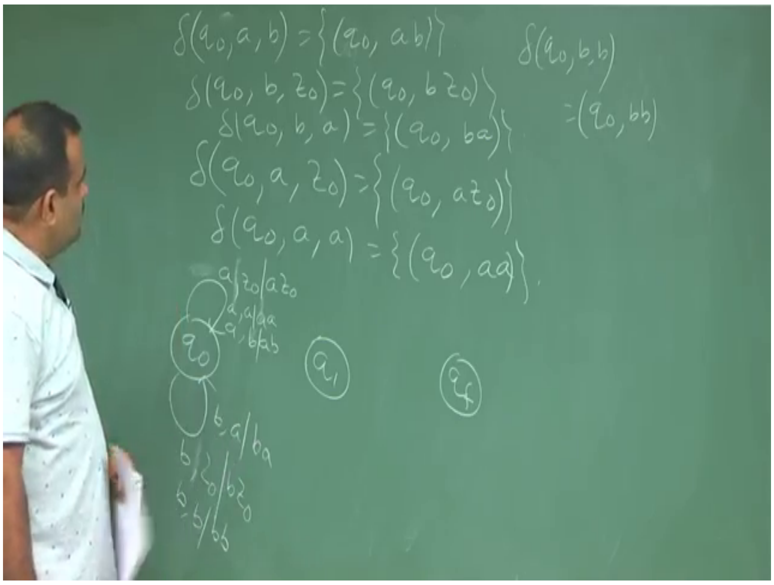


Suppose, we have a M like this q 0 and say q 1, q f these are the states and we have a, b, c these are the these are the sigma this is our q and we have a stack symbol. Stack symbol we are using a, b, z 0 and we have corresponding delta you have to define delta and then q 0 is there z 0 and f; f is q f basically f is q f. So, this is z 0 and you have q f over here. So, this is the states.

And, if we define the delta like this delta of say q 0, a, z 0 this is the singleton set q 0, a, z 0 and delta of q 0, a; a is equal to we can write this in a graphical form. So, this is q 0 and we have q 1 and then q f. So, we define delta of q 0 a. So, if we are q 0 a along with, if we have a move if the stack symbol is say z 0 z 0, sorry stack symbol here is z 0. So, this we underwrite. So, stack symbol is z 0 input is a and we are at q 0. So, this will be replaced by. So, we are going we are remain at q 0 and the stack symbol will be a z 0. So, this is the way we do that.

And, delta of q 0 a, so, a if we have a and if the stack is also a then this is going to replace by q 0 aa. So, this will be q 0, a, aa like this ok. So, only one move is there. So, yeah so, let us write this here.

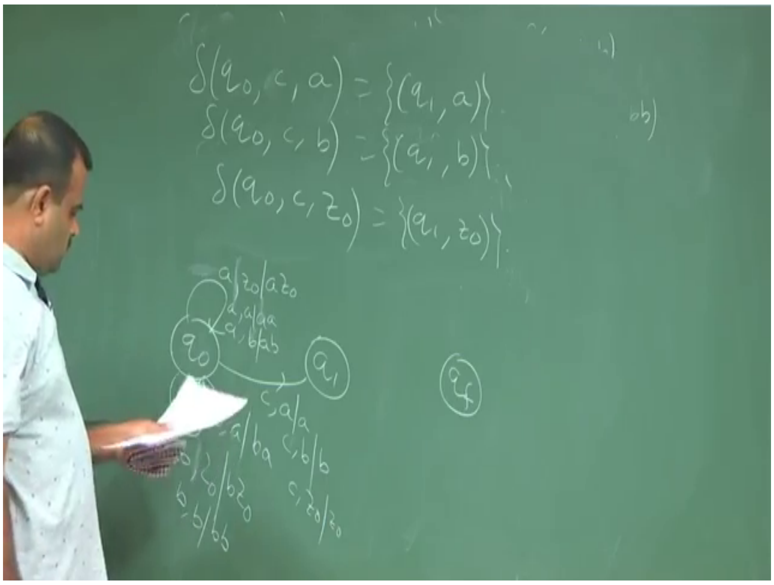
(Refer Slide Time: 25:33)



Now, delta of aa, so, now delta of q 0, a, b. So, q 0, a, b. Suppose it is again at q 0; so, this is also singleton set q 0, a, b. So, if we q 0, if we have a input a and if the stack is b then the stack will be replaced by q 0 q 0, a, b, a a is the stack so, q 0, ab. So, this will be replaced by q 0 will be remain same, stack will be ab like this and then if we have from q 0 to delta of q 0, b, z 0. This is we are going to have q 0 b z 0. So, this also we can write here. If we have b, sorry if we have b and if stack is sorry, if stack is z 0 then the stack is going to replace by b, z 0. This is about and we have one more and delta of q 0, b, a this is going to replace by q 0, ba.

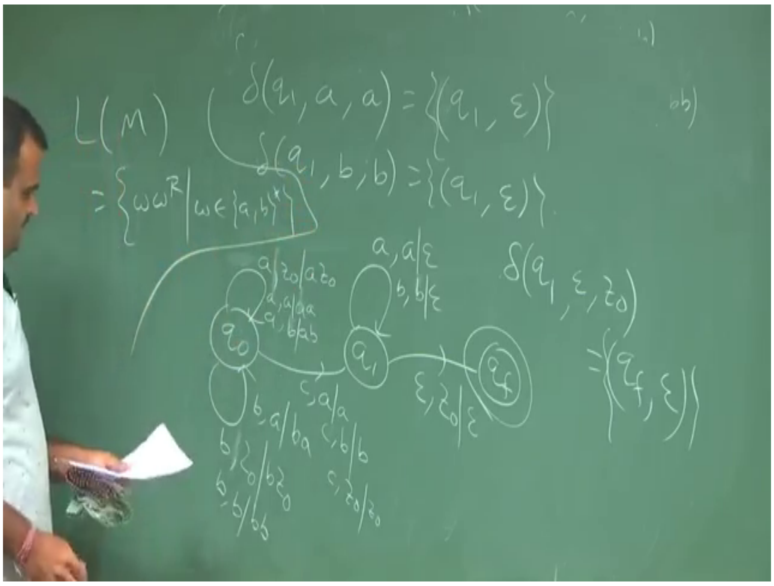
So, if we have a b and the in if the stack is a then it will be ba and it will be remain at q 0 and q 0, b, b; q 0, b, b so, this will be will be remain at q 0, and this is bb. So, q 0 will be remain same. So, b, b so, b and b so, will be bb. This is all rules of q 0, ok. So, now, with q 1 with c we are going to q 1. So, let me write the rules for c. So, let me erase these are the rules from q 0 with a, b.

(Refer Slide Time: 28:08)



Now, let me write the rules for q 0 with c. So, q 0, c small c, a this is going to q 1, a. So, that means, if you have a input c and the stack is a then stack will be remain at a, ok. And, again similarly delta of q 0, c, b this is also same as q 1, b. So, the meaning is so, if we have a c and if the stack is b it will be remain at b and similarly q delta of q 0, c, z 0. This will be again q 1, z 0. So, this is c, if it is z 0 we are going to q 1 and it will be remain at z 0, ok. So, this is the rules with q 1 and c. Now, we have a rule form q so from q 0. Now, we have a rule for q 1.

(Refer Slide Time: 29:34)



So, from q 1 delta of q 1 if we see if see if a and a is the input, so this will be say delta of which is basically q 1 epsilon ok. So, that means, if we have a q 1 and if we see a a will be remain at q 1. So, we have a over here. So, if we see a a and the stack is a then this a will be erase and q 1 if we see a b and if the stack symbol is b we are going to erase that. If you see a b and the stack symbol is b we are going to erase this ok.

And, from q 1 if we see a see yeah or epsilon this is the final step from q 1 if we see a z 0 we can go to q f epsilon. So, that is the final step. So, from here with epsilon and if we see a z 0 in the stack so, this will be empty ok. So, this is the rule and this is the final state this is the final state. So, this is one example of deterministic PDA. So, which language it is accepting? It is accepting by the final state it is accepting ww R; all the (Refer Time: 31:29) w is belongs to a, b star. This one can easily verify ok.

Thank you.