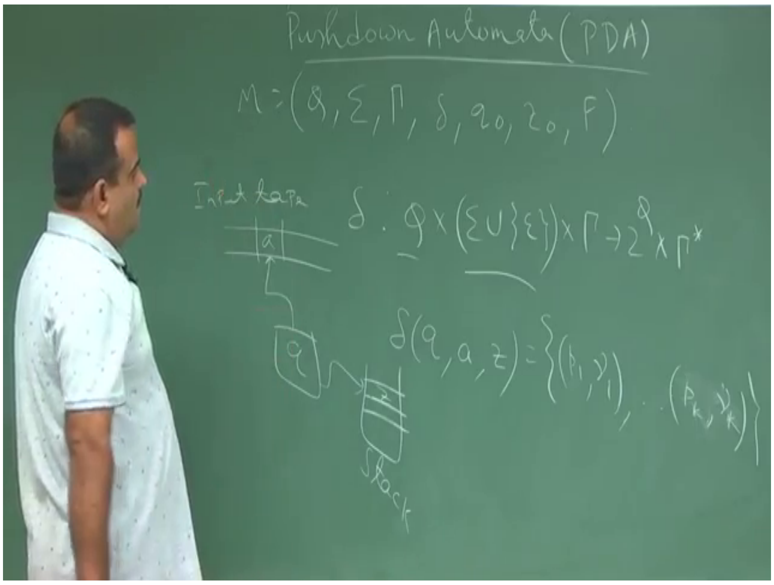
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 52
Equivalence of Language Accepted

So, we are talking about pushdown automata. We have discussed the, what do you mean by acceptance, language accepted by pushdown automata.

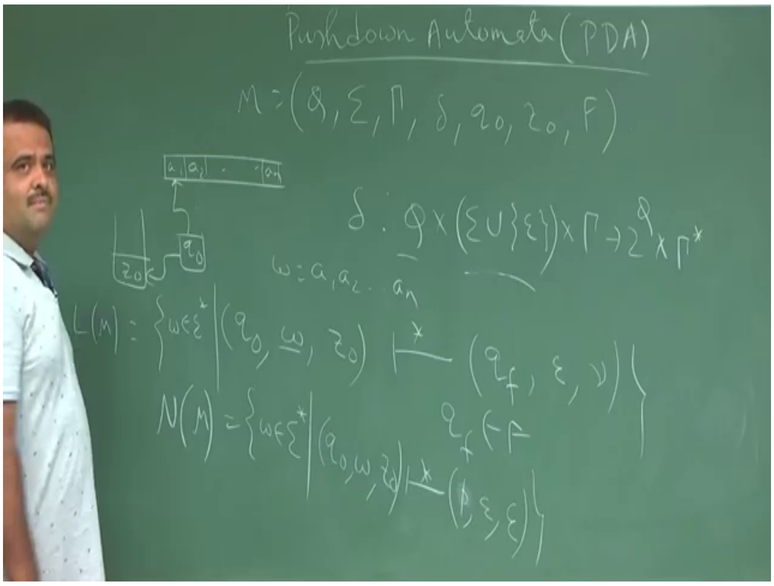
(Refer Slide Time: 00:29)



So, suppose we have given a pushdown automata M which is Q, delta, q 0 and z 0 then F. We know all these terms the Q is the set of states and this is the say input alphabet which is written in the tape and this is the symbol, stack symbols. These are all finite and this is the delta; delta is a function from Q cross sigma we do have a epsilon move, this is the NFA non deterministic PDA and you have a stack symbol and it is going to some 2 to the power Q and the stack symbol.

So, that means, given a so suppose this is our a and if we are at state q so delta of q comma a comma z. So, z is the top of the stack. So, top of the stack and input tape this is the input tape and this is the stack and stack symbol is initialized by z 0 and initial starting in state is q 0. Now, depending on this current state and the input it could be epsilon also, input alphabet and the top of the stack symbol it will go to some of the states. Since it is a epsilon non deterministic move so, we may have many options I mean. So, it can go to some p 1, gamma 1 dot dot dot p k sorry p k gamma k like this ok. So, this is the non and then we have seen the we have defined the ID Instantaneous Description.

(Refer Slide Time: 02:36)



So, if we have to if the input string is w which is say a 1, a 2, a n then the initial IDs initially our state is q 0 and w is the input. So, this is the tape; tape is a 1, a 2, a n which you have to read and initially it is q 0 and top of the stack is z 0, this is the stack. So, stack is pointing here and this is the situation this is the initial ID. So, w and then we have z 0.

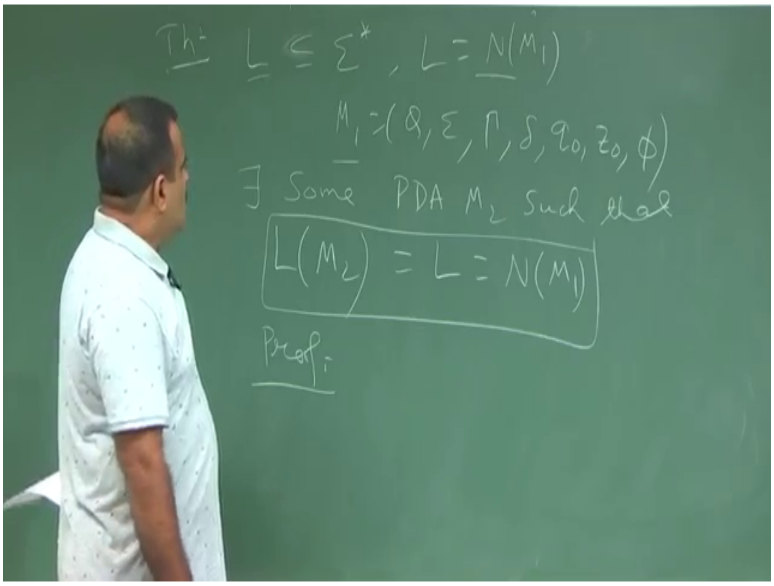
And if the several iteration we know the move relation depending on the delta we have a next state and next. So, this stack will be the z 0 will be replaced by. So, these way if you continue and if this is going to in the move relation maybe one or more relations more application of the transitions rule if this is going to some q f and epsilon and we do not care about this final. So, this set of w which is having this where q f is some final state then we called this is the language accepted by final state. So, this is we defined as L of M this is language accepted by the final state.

Similarly, language accepted by the empty string sorry, empty stack we have different like this is set of all strings which is starting with this and which is going to empty stack I mean here we do not care about the final state. So, empty stack means so, q so, we do not care about which state we are in p, but epsilon we read we exhaust the tape and then that stack must be empty. Then this is called the string accepted by w is the string accepted by the empty stack and this collection of all such string is denoted by the language which is accepted by the empty stack.

Now, we will see the equivalence between these two. I mean the they are basically equivalent language accepted if we have a PDA which is giving a language accepted by the empty stack then we have a corresponding PDA we can construct which will give us the language which will accept the same language accepted by the means of final state. And then we will see these are basically giving us the context free language, CFL ok.

So, let us first show the equivalency between these two acceptance. So, first we will show the language accepted by a empty stack if same as language accepted by a PDA. So, before that let us take a example we will take the example together with that.

(Refer Slide Time: 06:02)



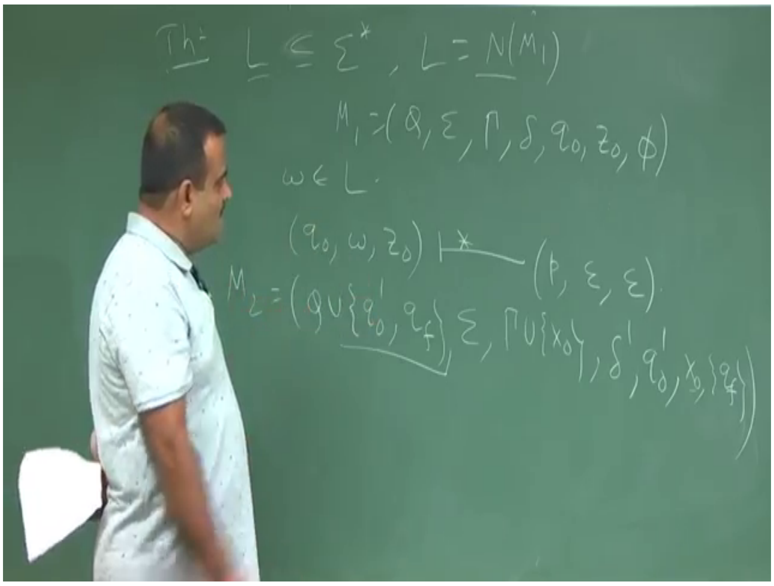
So, this is telling let L be a language L is a subset of L be a language such that L is; that means, we have a L is say language accepted by empty stack corresponding to the PDA M 1 where M 1 is some Q 1. So, this will be same because we are accepting the language. So, Q the sigma input alphabet we are not changing.

So, let me write the M 1 here, state maybe we can change M 1 or for M we can just have the no subscription sigma delta and say q 0, z 0 and we can say phi because this is the M 1 is the language accepted by the empty stack. So, we do not really care about the final states. There may not be any final state because this is the language accepted by the empty stack ok. After reading the string we must, I mean our stack should be empty that is the language accepted by the empty stack ok.

So, now, suppose this is accepting this language L which is a collection of the string coming from the input alphabet. Now, you have to construct M 2 then there exist some M 2 this is the theorem from M 2 such that M 2 the PDA exist some M 2 in a PDA pushdown automata M 2 such that the language accepted by this M 2 in the means of final state accepted by final state is same as this L which is same as M 1. So, basically we need to show they are exist M 2 which will be coming from this M 1 we need to construct M 2 and they should give us the same language, so that you have to prove.

So, let us try to prove that. Let us try to construct the M 2 given M 1. So, the idea is so, this M 1 is telling, M 1 is accepting all the string this is the proof sketch will give.

(Refer Slide Time: 08:58)



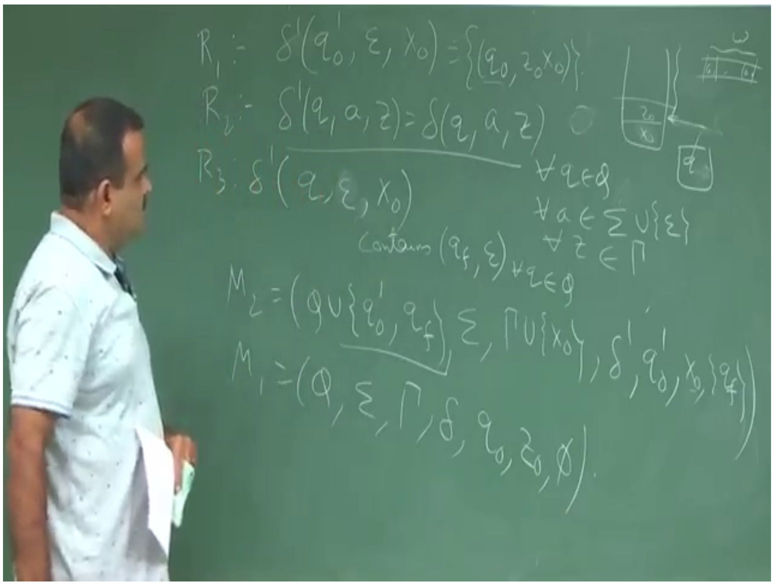
So, what is the idea? Idea is we are so, if w belongs to L then for this string w what we are doing we are just starting with this q 0 then w z 0 and it is end up with after this is the move relation. After repeated a repetition of application of this delta it will go to some state p and then epsilon and this is also epsilon ok, p is any state we do not have any final state over here. So, p is any state. So, that is the w.

Now we want to have construct a M 2 which will accept this w in the means of final state. So, for that what we do? Let us construct M 2. M 2 will be state will be we have to have we are taking 2 new states; one is for final state. So, q 0 prime and say q f this is the state and we have sigma and this we are using a another say stack symbol which is the X 0 that we are going to use for the initial stack symbol for this new DFA, for this new PDA ok.

And we are need to define the delta hat that will be the change delta and q 0 prime is the new final state and X 0 is the this one the new initial stack symbol and we have q f; q f is the final. So, this is our Q this is our M 2. So, M 2 consists of all the states of M 1 along with 2 new states and this q 0 we are q 0 prime we are going to use for the new starting state for this new automata and q f we are going to use for the final state this is our f.

And this will be remain same and here the stack symbol it will be all the stack symbol of the M 1 along with a new symbol X 0. So, X 0 we are going to use for the in a new initial stack symbol for this PDA and delta prime we need to different; so, how to different delta prime? So, let us try that. So, this I will come back.

(Refer Slide Time: 11:50)



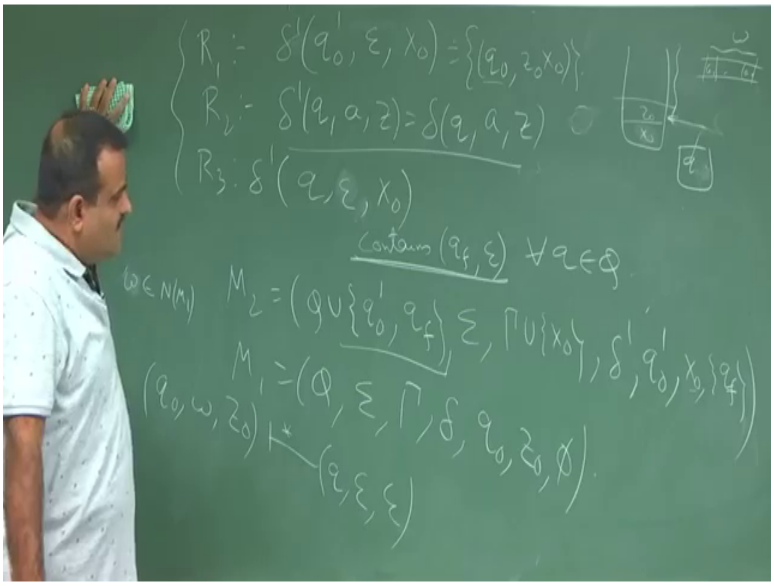
So, this is the this one. So, let us different delta prime. So, this is the row. So, delta prime of so, initially q 0 prime is the first one with the epsilon move and our initial stack symbol is z 0 sorry for M 2 for M 1 it is X 0, but for M 2 you want to put X 0 over there. So, initially we are at q 0 prime. With this what you want to do? We want to go to q 0, but with the new. So, this is nothing, but q 0; q 0 is nothing, but so, we have M 1. M 1 let me write the M 1 M 1 is q, delta, q 0, z 0, phi. So, this was the initial this.

So, q 0 is the state for the M starting state for M 1. So, q 0 along with so, what we do we replace this by X 0 z 0 we replace this by sorry z 0 X 0. So, now, it is z 0 and now we are at the initial state of M 1 and then we will follow the rule of M 1 ok, then we will follow the rule of M 1 transition rule of M 1, so, that is there. So, q, a, z which is same as delta of q; q is in this M 1 z and this is true for all q belongs to Q and all a belongs to it could be epsilon also and for all z it is the symbol coming from M 2.

So, this is basically rules of M 2. So, after we assign this then we will follow the rule we are at our state is that q 0 after we assign this now we are following the rules of M 1. And once we follow the rules of M 1 if w has to accept then w will the end of this is the if w is accepted string this is a 1, a 2, a n if this is w if w is the acceptance string so, it will end up with this will be empty. So, that means, X 0 will be there remaining and this does not matter this will go to any state and it will end up.

So, that string we have to accept in the sense of final state. So, that we are going to do now. So, this is the R 3. So, once we finish up this that reading a, once we finish up this epsilon and if we see a q 0 over X 0 over here; that means, this is gone and this is q. So these contains q 0 comma epsilon for all q belongs to Q, this is the rule.

(Refer Slide Time: 15:59)

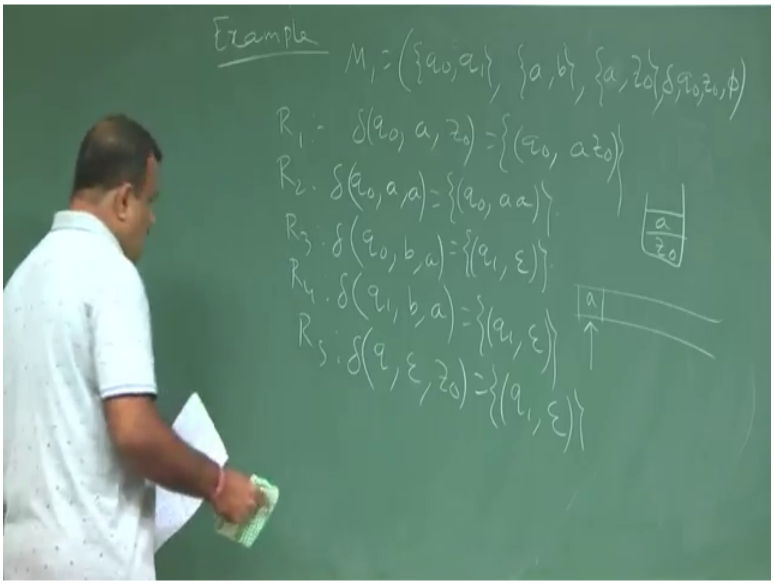


This is for all q belongs to Q. So; that means, if your w if we are starting with a w; if we are starting with a w so, q 0; q 0 is the starting state for this M 1, q 0 w and then z 0 and if w is belongs to L of M 1 I sorry N of M 1 then it will end up with what? It will end up with some state we do not care about that because this is accepted by the empty string and this will be empty and this will be empty.

Now, once this is empty so, it will be for new PDA it will be remain X 0 ok, then we have to put it to the final state so; that means, this should contain once we reach there we should have; we should have a move which will reached us to a final state of the new automata because it may happen that before that we may so halt. So, that is why it is containing; it contained this. So, it is not always it is unique like before that we it may halt; that means, then our stack may be empty before that then we have to halt there we have. So, this will, this branch will not end up with that because we may have many other variable to read there, may be other input to read there ok. So, this is the idea.

So, once we reach to this epsilon; that means, X 0 this will be vanish, then the X 0 will be remaining. So, then if our string is then we should have the option to go to the final state because if the string is empty then it must accept that string. So, that is why this is the rule this contained this ok. Even we could put here something like gamma because in this case we are only concerned we should reach to a final state. So, instead of epsilon we can put gamma over here anyway we will take an example then it will be more clear so, this is the idea. So, we introduce two new states along with a new stack symbol which is initialized in the stack and then we have a final step like this. So, let us take an example.

(Refer Slide Time: 18:56)



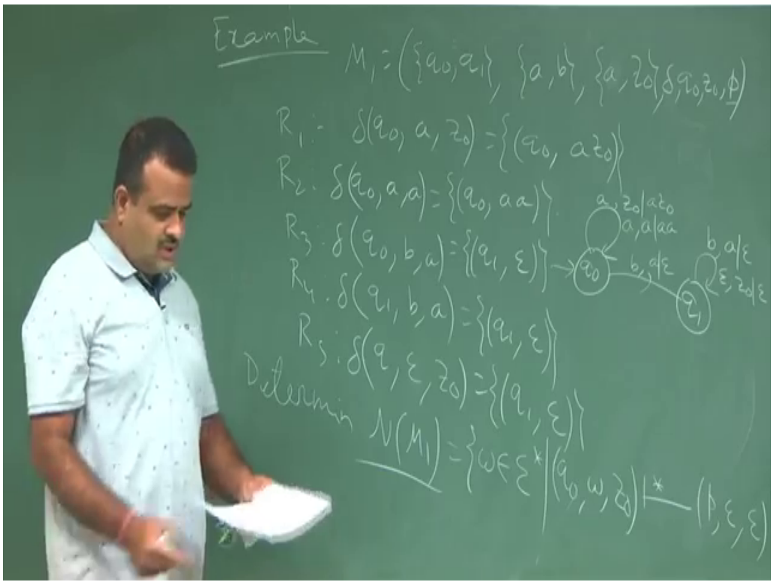
So, let us take a PDA which is accepting the some language. So, in the means of empty stack let us consider a PDA like this. So, this is q 0, q 1, a, b. We have only two stack symbol 0 and sorry is not 0, this is a; a and z 0 and we have delta q 0, z 0 and phi; q 0, z 0, phi ok.

Now, let us define the delta. So, delta is as follows; this is R in 1. So, this is we want to accept the language like a to the power n, b to the power n. So, delta of q 0, a so, once you have a. So, this is the starting one, once we have a what we will do? We will push a into the stack. So, this is belonging. So, we will be at q 0 and we push a into the stack, so like this. So, this is the sorry. So, this is the stack which is initialized by z 0.

Now, if the input tape is say like this a. So, what we do? We push this on the stack and when you see a b will pop it from the stack. So, this is one rule delta of q 0, a, a again will push this into the stack so, q 0 a a. R 3 delta of q 0, b, a. So, it will be it will erase it will pop a. So, that means, it will erase. So, this will go to q 1 and it will these are erase say q 4; so, now, q 1, b, a.

Now, if we see a a in q 1. So, again it will erase that, but it will be remain that q 1 and the last one is delta of q 1, epsilon, z 0. So, this will be q 1, epsilon ok. So, if you draw it will be like this; like draw this.

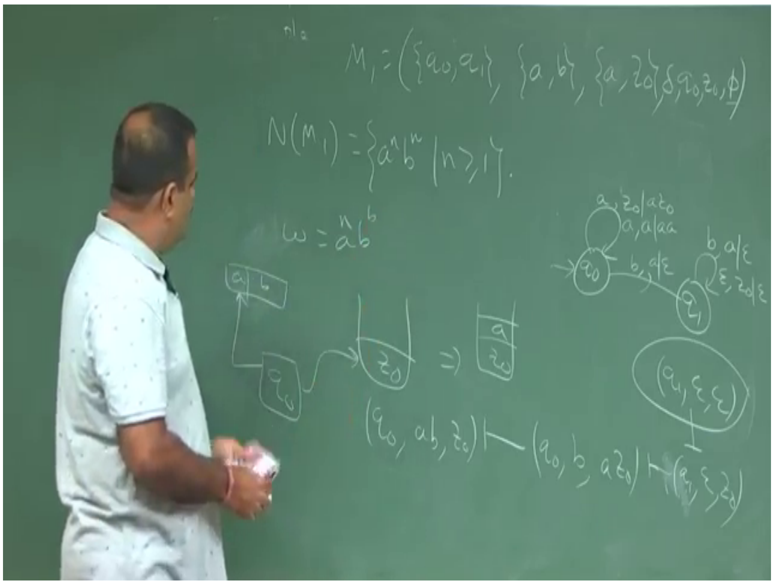
(Refer Slide Time: 21:57)



So, we have 2 states q 0 and q 1. This is the starting state and with q 0 if we see a and if z 0 is the stack symbol then it will be replaced by a, z 0 and if we see a a again and if the start stack symbol is a then it will be a a. So, we are pushing a and if we see a b then it will go here, if we see a b and the stack symbol is a then it will erase a. Then once we reach b like q 1, then if we see a b and if the stack is a it will erase a and if we see a epsilon there and if the stack is z 0 then it will erase z 0 ok. So, this is the transition functions of this.

Now, what it is accepting? So, this is by empty stack. So, that is why we do not have any final state over here, final state is empty. So, what it is accept accepting? So, how to determine? We have to determine N of M 1. So, language accepting so, this is the set of all w such that if you start with q 0 with this w with z 0 this must go to some of the state we do not care about that, but we must end up with reading the string and it must go to the it must erase all the symbol from the stack. So, we have to find this. So, let us find this. So, this is the automata we have.

(Refer Slide Time: 24:45)

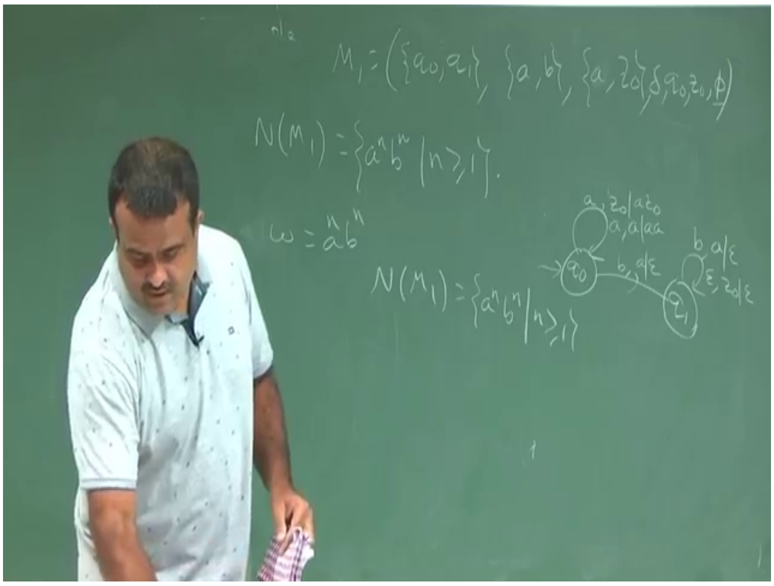


So, our claiming this will be yeah, we are claiming that this N of M 1 this will be a to the power n b to the power n alright, n is greater than 1 ok. So, this we have to show. So, for that what we can do? We can just. So, we have to show this that a it is just taking the a to the power n b to the power n. So, basically the idea is. So, whenever we see a one so; that means, if we start with say a a ab say. So, if we are q 0. So, this is the in suppose our w is ab. So, what we do, we just put a stack symbol z 0 and our initial this is q 0 and our tape is ab.

Now, what is the next step? Now, we are seeing a and stack is z 0. So, it will be going to so, this ID. So, q 0, then ab and z 0, so, this is going to have this is going to. So, the stack will be like this, so, it will be z 0, a. So, if we see a a in the input it will be it will push a symbol there and if we see a b then it will pop. So, that means, the because you have to count the number of a’s number of b’s. So, that is the way we can count. So, then it will go to again q 0 and the stack symbol is this is the b, b you where to read and this z 0 will be replaced by a z 0.

Now, once we see a b then it will be q 0 to b. So, if it is say b it is going to q 1. So, q 1 so, this will go to q 1 and this is empty and we have to pop erase a. So, that is z 0. Now, in here if we see a z 0 this then it will just make it empty. So, it is q 1 and less empty and q. So, this is the accepted by the final state. So, this is accepting ab. So, in general we can, so, this is accept a to the power n b to the power n because if we just take a to the power n b to the power n it will just keep on pushing a until it c ab once is c ab then it will keep on popping a.

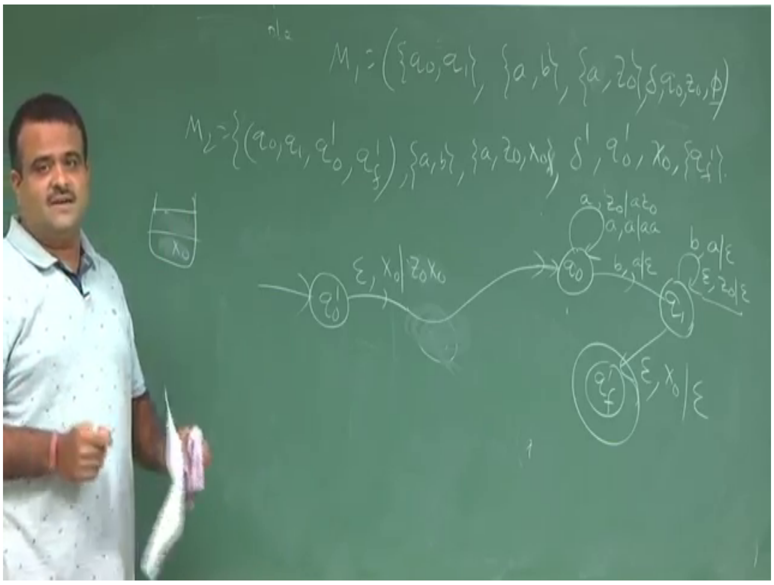
(Refer Slide Time: 27:38)



And so, it is a count like how many times a is occur, how many times b is occur. So, when it will have a same number of push and pop then it will be this b will vanish and it will be here and we will see a z 0 in the final state and then. So, eventually this is N of M 1 is nothing, but a to the power n, b to the power n; n is greater than equal to 1 ok. So, this is the language accepted by empty stack.

So, now, we will have the corresponding language which is accepted by this final state. So, for that we can take this. So, these we have to add a new symbol this we can erase.

(Refer Slide Time: 28:25)



So, we have to find M 2; for M 2 what we are doing we are keeping the all the states of M 1. So, all the states of M 1 along with the new starting state and the new final state and this will be remain same and we have a X 0 is the new initial symbol in the stack and this is the delta hat and q 0 prime, X 0 and this is q f prime.

Now, the rules is like this. So, this is the starting for M 2. We start with q 0 prime and with the epsilon move with the epsilon and we see a z X 0 in the stack because X 0 is the initial symbol in the stack for M 2. So, this will be going to X 0 sorry, z 0 X 0. Now, we are here we are at q 1. So, now, it will be replaced by z 0 X 0.

Now, z 0 is the starting symbol for M 1. So, now, we have a rule for M 1. So, we have a rule for M 1. So, this is one. So, this will go here. So, we have a rule for M 1 like this and then after here one, so, this will be. So, this will keep on doing this. So, this is this will change like this. This will be X 0 so, epsilon of. So, from here then we have a q f prime which is nothing, but if we see a epsilon move and if we see a X 0 in the m in the stack then it will erase epsilon. So, this is the final state ok.

So, once so, after this we are playing with M 1. Now, if the w is accepted the string then that w will give us erase this and it will reach to X 0 and the input tape is finished, epsilon and then once we reach that; so, that is the situation over here, once we reach that then we have to have another move to the; to go to the final state because you want to accept this as a accepting by the final state. So, this is the one.

So, in the next class we will see the converse that the acceptance if given a NFA which is accepted by the final state we can have corresponding, we can construct a corresponding sorry given a PDA which is accepted by the final state we can construct a corresponding PDA which will be accepted by the empty stack. And then we will see the equivalence between the context free language and the PDA. So, context free language will corresponding to the PDA with the empty stack.

Thank you.