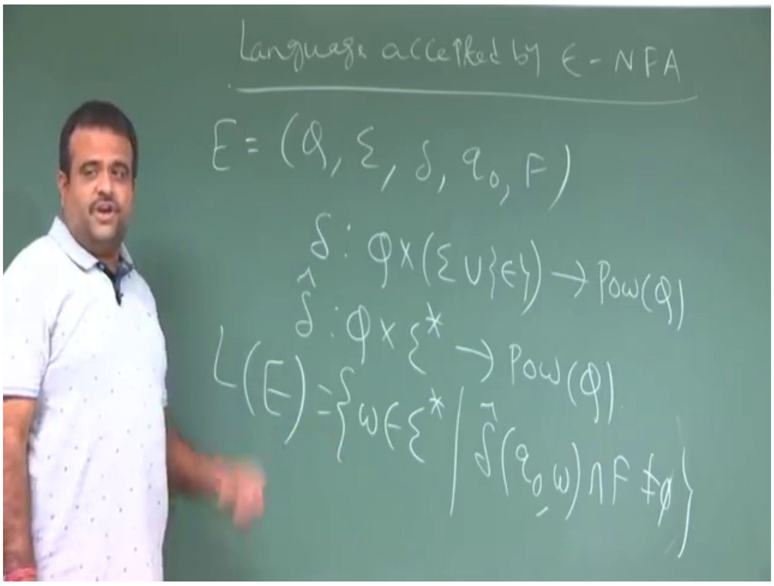
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 13
Language of Epsilon-NFA

So, we are talking about Epsilon NFA.

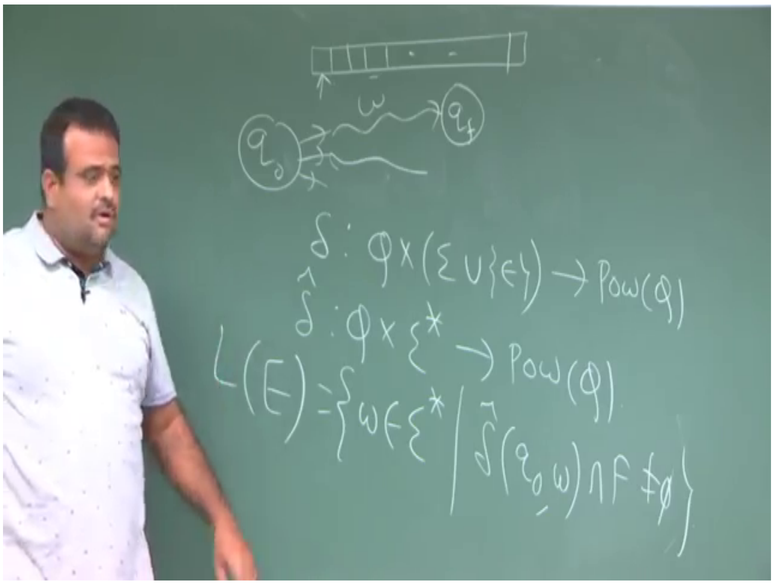
(Refer Slide Time: 00:19)



So, just to formally define the language accepted by epsilon NFA. So, suppose we have a epsilon NFA Q, delta, q 0 F. So, this we know this is set of all states, which are finite set this is the set of all possible alphabet and this is the starting state and this is the final state and this is the transition rule, but here we have a this is this can take only one alphabet; so, or the empty string. This will go to the power set of Q ok.

And then we have extended this on sigma star to the power set of Q and then we define the language. So, if this is E, then the language of E the epsilon NFA is nothing, but set of all w belongs to sigma star such that delta hat of q 0 and w did its intersection is nonempty. So, then this intersection is not empty. So, this is a collection of all string which is racing you reaching to the accepted state. We start with the we start with the starting state q 0.

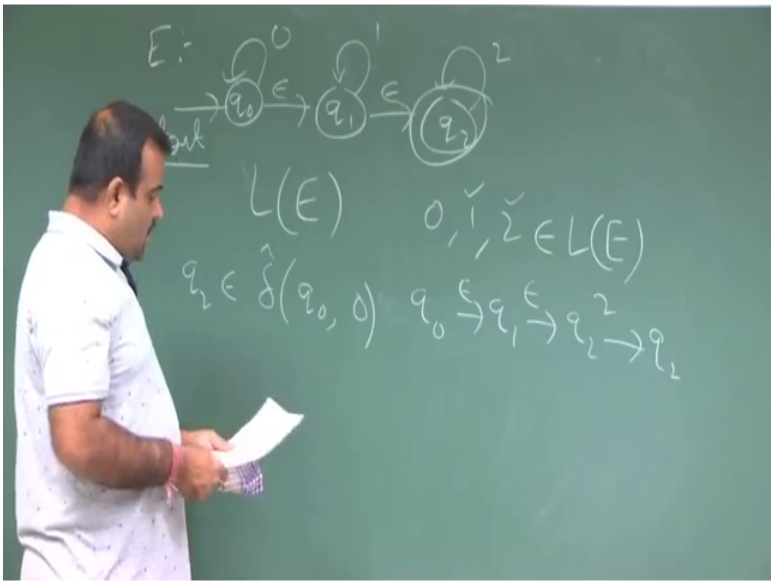
(Refer Slide Time: 02:11)



So, we have a string w, this is a collection of alphabets and we start we have a suppose you have a tape like this it is just we start with q 0. And since it is epsilon NFA, we have epsilon move we have moved to the non deterministic move, we have moved to the many states.

So finally, if some of the path we can reach to the one of the final state, then we say with this w along with the epsilon move. Then we say this w is the string which is accepted by this epsilon NFA, and the collection of such string is called the language by the language of that NFA. So, let us take two quick example.

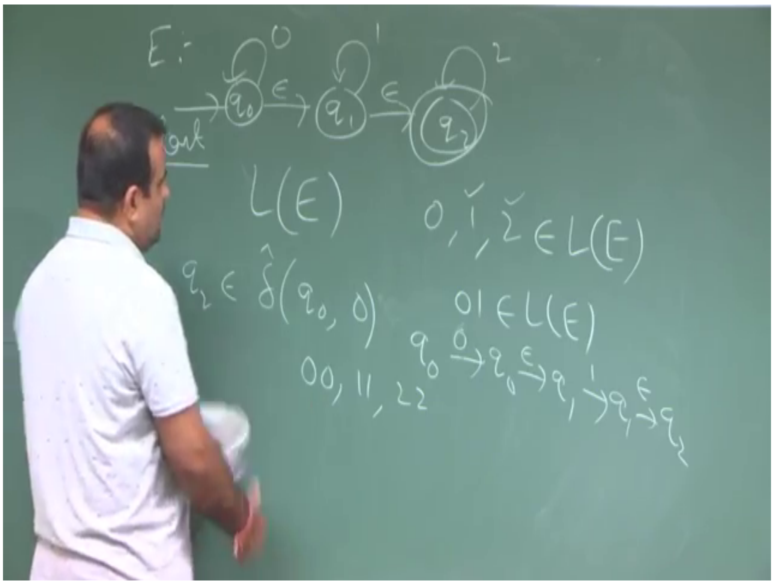
(Refer Slide Time: 03:08)



So, we have seen this example before q 0 we have 3 state q 1, q 2, q 0 epsilon 1, epsilon 2 and this is the final state and this is the starting state or start state. And so, this is our E epsilon NFA. So now, we want to get the L of E the language accepted by this. Now, 0, 1, 2 all are belongs to L of E because any one of this. So, 0 if we start we can so, from 0 this is delta hat of q 0 0, this we can reach to the this is this q 2 is a subset of this q 2 belongs to this why because, we can start with q 0 and with 1 0 we can be at q 0, then we can move to epsilon move q 1 epsilon move q 2 final state. So, we are reaching to the final state.

Similarly, for 1, for 1 we can execute the sequence from q 0 we can have epsilon move, q 1 with epsilon move again at q 1 and then epsilon move at q 2. So, final state so, 1 is belongs to 1 is a accepted string. Similarly, 2 also accepted string we can just move to the from q 0 with epsilon we can go to q 1, with epsilon we can go to q 2, and then we 2 we can go to q 3 sorry q 0 and with 2 we can remain at q 2. So, this is the final state.

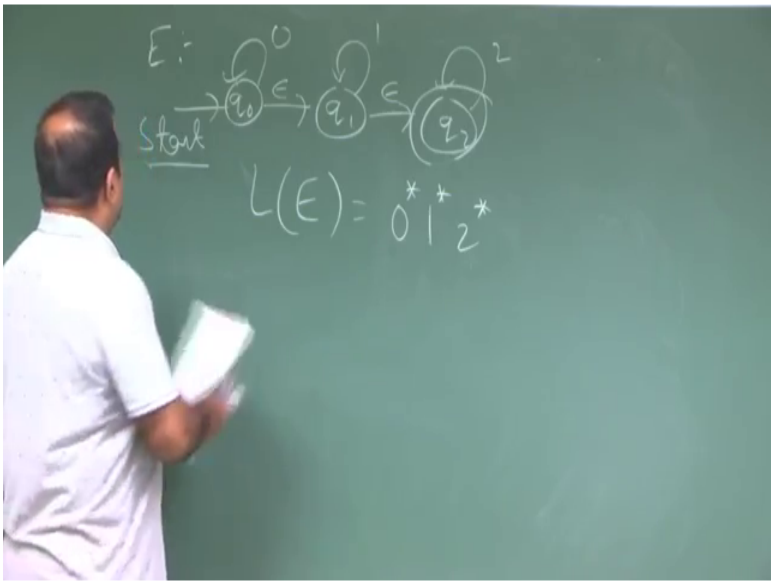
(Refer Slide Time: 05:13)



So, this is the even 0 1 0 1 is will also belongs to L of E because if we take 0 1, we can start from q 0 with 0 you can go to q 1, q 0 again and from q 1 we can go to q 1 from q 0 with epsilon move, then with one we can go to q 1 again then with epsilon move q 2 ok.

So, this is the way we can figure out 00 11 22 like this any combination of this. So, all are reaching to a this. So, this in fact, this language is all string with any number of 0's, followed by any number of ones, followed by any number of twos.

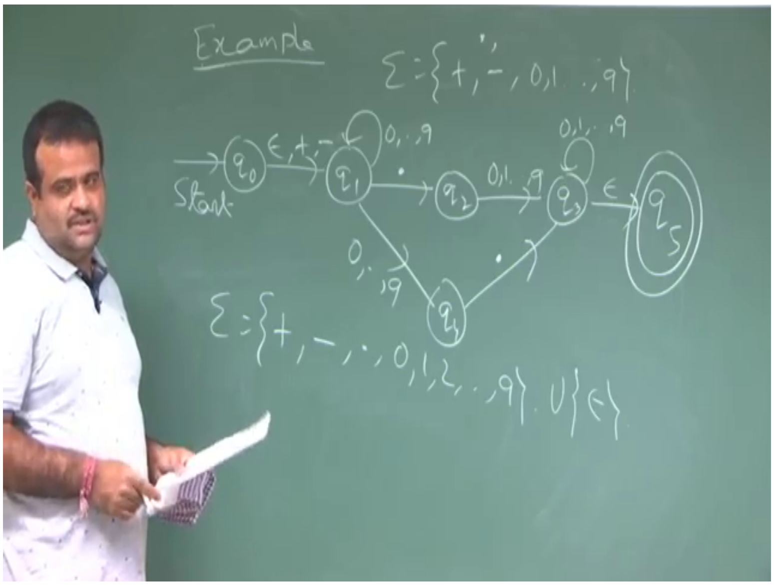
(Refer Slide Time: 06:07)



So, this is w w content w is basically this is actually denoted by 0 star, 1 star 2 star will defined this regular expression in next after two lecture.

So, this is this is basically denoting the any number of 0’s, then followed by any number of 1’s then followed by any number of 2’s. So, this is the way we defined this. We take another example we take another example.

(Refer Slide Time: 06:56)

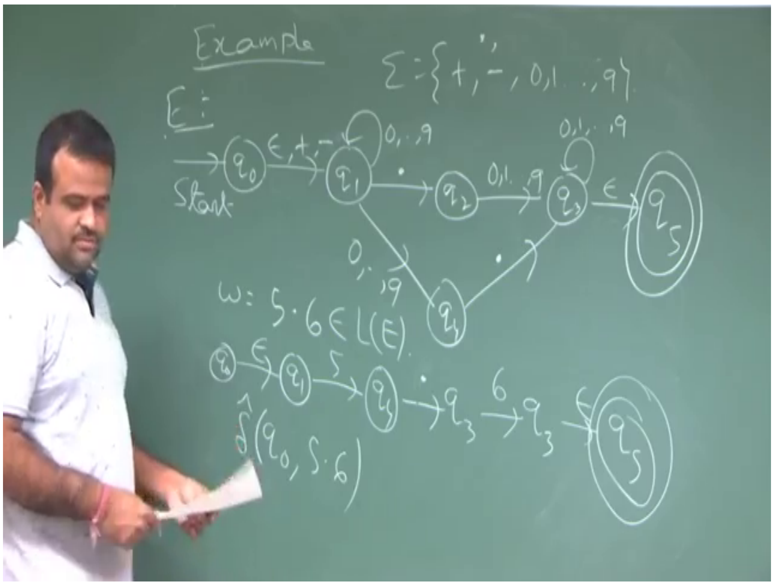


So, here suppose we have we are taking this as a plus dot suppose arithmetic operation we are doing, and we have all the digit up to 9. So, this is our sigma and who is what is our. So, this is the starting state and we can go from here to q 1 with a move either epsilon or plus or dot and we have a minus also yeah. So, minus we have a dot also minus so, minus also. These are the arithmetic operation you know this is sorry this is minus is here dot is not there ok.

Now, from q 1 will be remain at q 1 with any of the input as digits. So, 0 1 to 9 and from q 2, q 1 if we go to q 2 if you see a dot as a input and here q 4 if we see a digit or any of the digit 1 2 0 to 9. And then from q 2 again if we see a digit 0 1 2 9 we go to q 3. And from q 4 if we see a dot, you go to q 3. And from q 3 if we see a another digit we will be at q 3 and then by the epsilon move we can go to q 5 and this is as the final state this is our final state. So, our sigma is plus these are the operation and we have 0 1 2 up to 9 all the digits, and we have a epsilon move. So, with the epsilon move we can we have this operations ok.

Now, for this we want to find the language, we want to see what are the string accepted by this. So, we can see only the epsilon is not accepted by this string, because if we have epsilon that was we can go to q 1, but we have to reach to q 5 in order to have a string to be accepted ok. So, we will see what are the string will be accepted by this epsilon NFA ok.

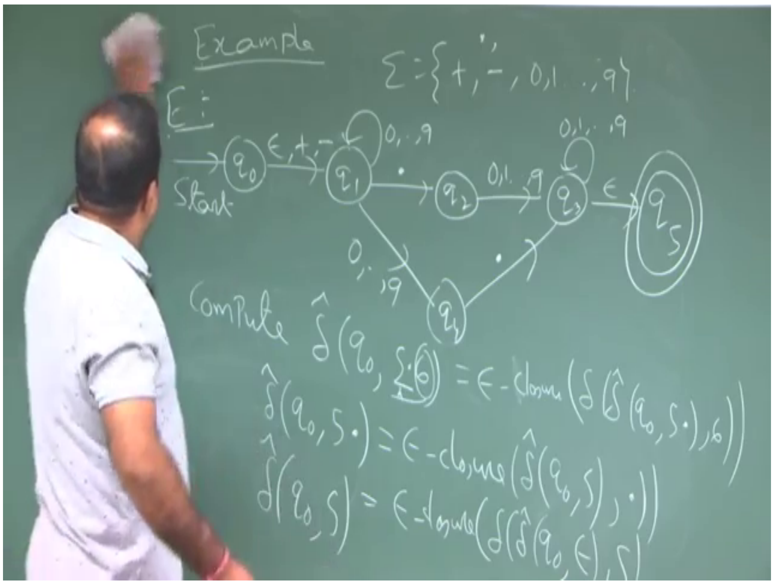
(Refer Slide Time: 10:06)



Now, we want to see 5 dot 6 this is our w say 5 is coming from here these. So, these are all string of alphabets 5 dot 6. So, what we do? We start with q 0, then we can go to with epsilon move we can go to q 1 then from q 1, again we can yeah we can go to q 1, and then from q 1 we can go to we see a 5. So, this is epsilon we see a 5 over here then we can go to q 4. Because q 4 we can go by seeing any digit in the tape from q 4 we see a dot with dot we can go to q 3 this is one of the path, there are many other path are there. So, we will we will construct those. So, we can go to q 3 by saying dot and then from q 3, we can hop at q 3 by seeing this 6 q 3 and then with the epsilon move we can go to q 5 which is the final step.

So, this is accepted. So, this is this belongs to L of E this is the accepted string. So, because this is giving a one path there are other path this is not this is not delta hat q 0 this. So, this is one path there are many other paths. So, what are for those we have to find the epsilon closure of these we can try with that yeah.

(Refer Slide Time: 12:17)

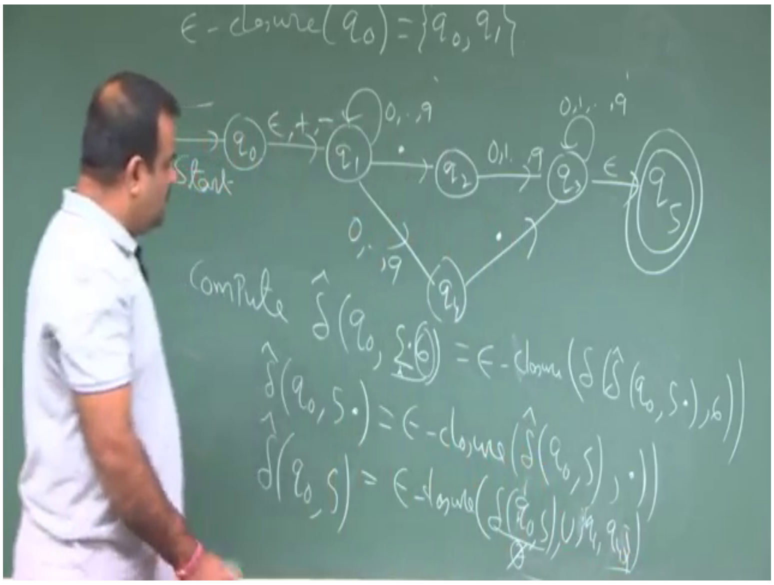


So, suppose you want to compute delta hat of q 0 5 comma 6 5 dot 6 this is a string, and we just shown in the one path it is accepting, but what are the other path i mean like if we have to compute this, this will be a set.

So, to find this, this is nothing, but epsilon closure of closure of the set delta of delta hat of q 0 5 dot comma 6 ok. Because this is the we remember x a this is our a, this is x both together. So, this again we can write as 5 comma dot we write 5 comma dot delta hat of delta hat of q 0 5 comma dot, this also we can write epsilon closure of delta of this is a set q 0 5 comma dot delta hat of.

Now, again to find this is delta hat. So, it is not delta; delta hat involves the epsilon move also. So, delta hat of q 0 5 it is nothing, but epsilon closure of delta of delta hat of q 0 epsilon comma 5 ok. I take this place now we can get this, this is a epsilon closure of q 0. So, what is the epsilon closure of q 0?

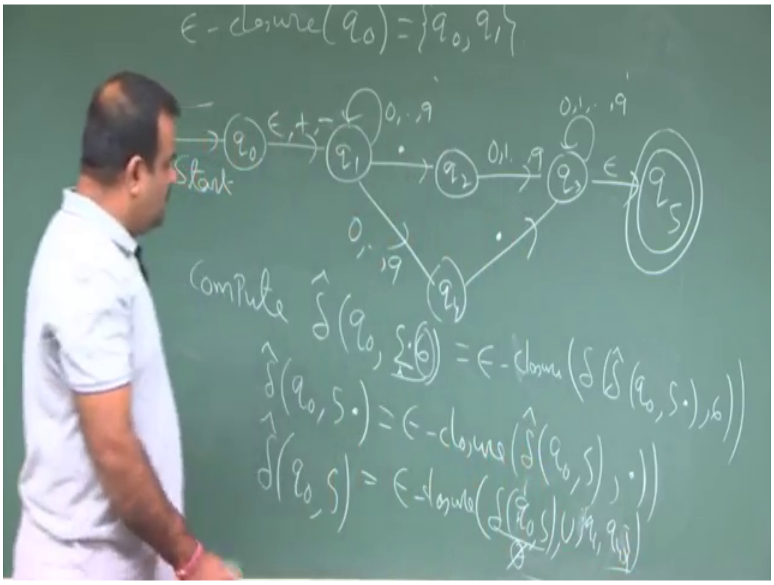
(Refer Slide Time: 14:29)



So, epsilon closure of q 0 is nothing, but only q 0 and q 1 because we can with epsilon move we can go to q 1 or we can remade at q 0 ok. So, this set is nothing, but q 0 and q 1. So, from q 0 q 1 with 5 we can go from where to where. So, from q 0 q 1 we can go to. So, this we can write as this we can write as, this we can write as q 0 q 1 that calculation we will do now ok.

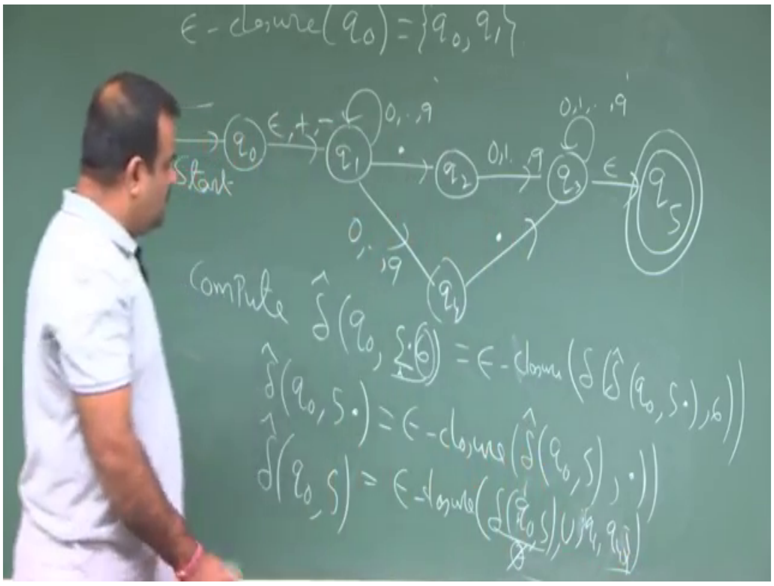
Now, this is basically epsilon closure of q 0 and q 0 comma 5 epsilon closure of q 0 comma 5 and union of delta of q 0 q 1 comma 5. So, this will give us q 0 comma 5 is we know how to move for. So, from q 1 comma 5 we can go to this is empty. So, because from q 0 there is no move with the digit, but this one is q 0 comma 5; from q 0 comma 5 we can go to q 0 sorry q 1 comma 5 q 1 and q 4. So, this is this set is nothing, but q 1 comma q 4. So, this will give us q 1 comma q q 4. So, this set is q 1 comma q 4.

(Refer Slide Time: 16:08)



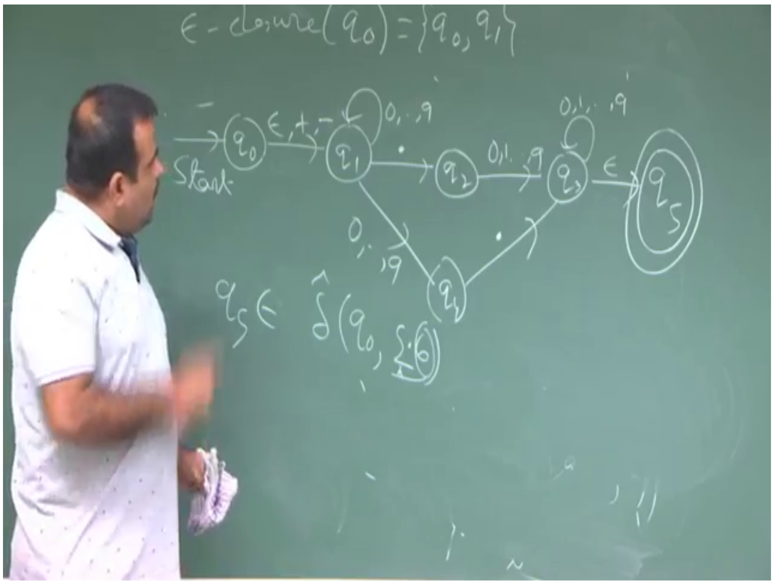
So, this we put it here, this is the epsilon closure of delta of. So, q 1 comma q 4 so, if you continue like this will be getting this to be this to be eventually q 1 and q 4.

(Refer Slide Time: 16:35)



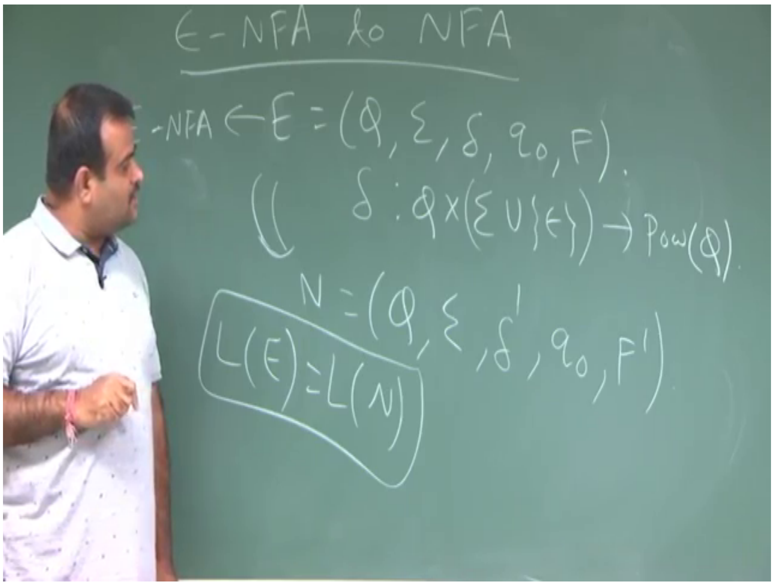
Ah q 1 q 4 and if you continue like this will be getting q 0, q 4 and finally, we will get q 5 also here.

(Refer Slide Time: 16:58)



So, this q 5 will be subset of this, if you continue like this. So, from here we can say this is otherwise we can just compute whole string, that will be given in the lecture not ok. So, now we want to see how we can given the NFA epsilon NFA how we can construct a equivalent NFA. So, that is our next target.

(Refer Slide Time: 17:27)

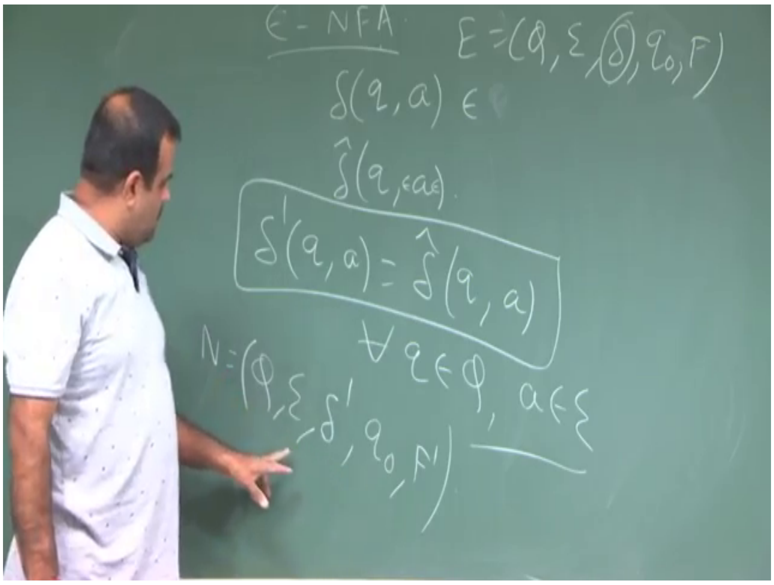


So, epsilon NFA to NFA, that we want to see how we can construct a epsilon NFA to NFA. So, suppose we have given a epsilon NFA M or E q 0 F ok.

So, now here delta is we can have yeah delta is basically a function from Q cross power setup Q. Now from here we have to construct a this is epsilon NFA ok. So, now, from here we have to construct a ma NFA N such that with the same state, state will be same input alphabet will be same, only thing will change this delta to delta yeah. So, q 0 will be same and F prime we need to change F prime will change; you have to construct this such that L of ok. So, they are accepting the same language this we have to construct

So, how we will construct this delta prime? Delta to construct the delta by basically NFA has no epsilon move. So, we can just add the epsilon closure. So, basically we are going to add epsilon closure of the state that is all. So, let us define this.

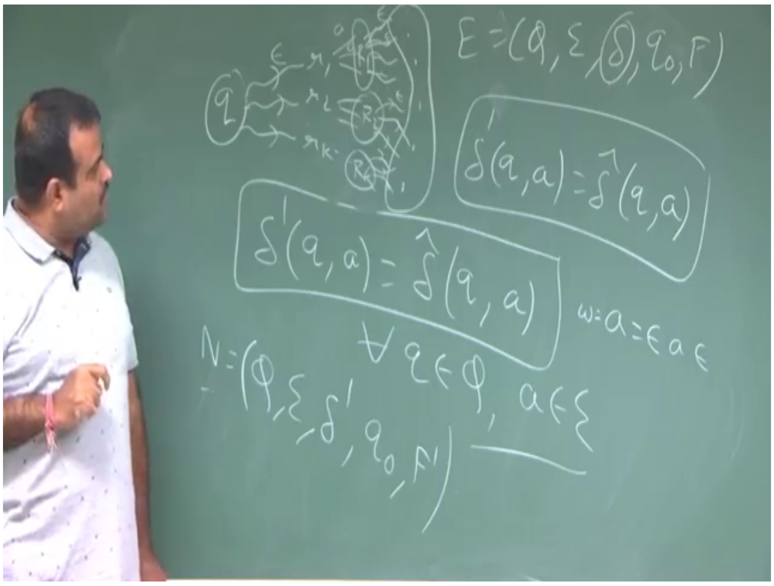
(Refer Slide Time: 19:42)



So, for epsilon NFA what we have? We have a delta we start with q I mean from a state, and this is this is a member of this is a power set of this is some states. Now there is a difference between these and this because here we are just defining the we are we are including the epsilon closure of this from both the side. So, these will be our delta prime. So, delta prime is nothing, but delta prime of q comma a is basically delta hat of q comma a and this is true for all q belongs to q and all a belongs to sigma. This is the way how we define the delta hat this delta hat is the NFA, this or NFA is transition rule is delta hat q 0 is same only F prime we need to defined and delta has this define.

So; that means, we are just from q 0 we are taking all the epsilon closure and we are reaching there and from there we take a move with this delta, with the input a and then we reach to the another sets of states, and from there again there is option to take the epsilon move. So, we include all of these. So, that is nothing, but our delta hat, because we have seen there is a difference between. So, our epsilon NFA is Q, sigma delta q 0 F. Now, this one this one is with a we can define these as on a and epsilon, but there is no movement of epsilon here ok. So, what is the so, basically we how is this different?

(Refer Slide Time: 21:56)

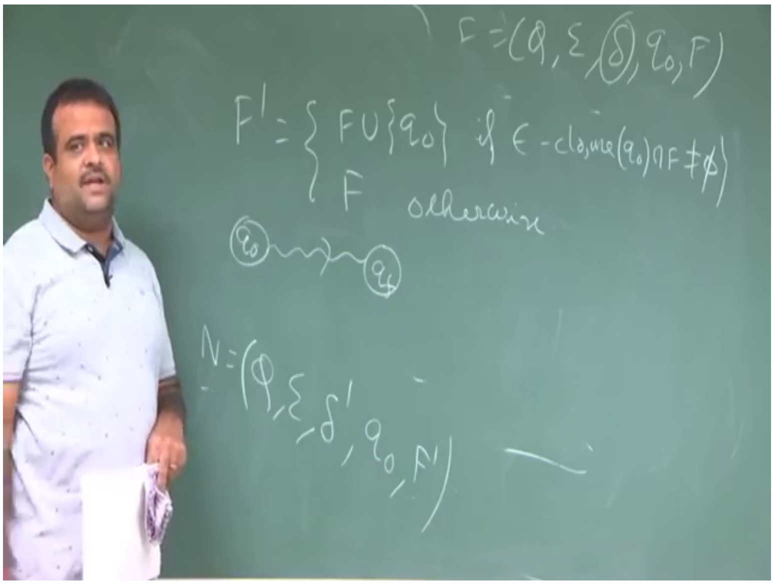


So, we are at q, now we consider the epsilon closure of this so; that means, if all the path by only epsilon arc we can go say r 1 r 2 say r k then from r 1 r 2 r 1 by a we can go to some of the R 1, then for with r 2 also we can go to R 2 r k R k like this. So, these are the set of states.

Now, again from this set of the state we can go by epsilon move. So, this is our all epsilon move, these are all epsilon move. So, these are all basically our these states are basically nothing, but delta prime of q comma a delta prime is this one. And this is nothing, but we know delta hat of q comma a because for delta hat we are we are accepting a alphabet as a string. So, string is having a if we consider w, a epsilon a epsilon ok.

So, before that we can have a epsilon move and after that also we can have a epsilon move. So, this is the string. So, this is the way now what is F prime? Now what is F prime?

(Refer Slide Time: 23:47)



So, F prime is nothing, but F we include q 0 if epsilon closure of q 0 and F is not empty otherwise it is F because with the epsilon move only we can if we are. So, we are at q 0. So, with the epsilon move if we can reach to the any one of the final state of F, then we have to add q 0 because that will come under this F prime otherwise it is F prime, if there is no epsilon closure of q 0 belongs to the contain the final state. So, this is the way we construct epsilon NFA to NFA. So, in the next class we will see in we will discuss two example of such construction.

Thank you.