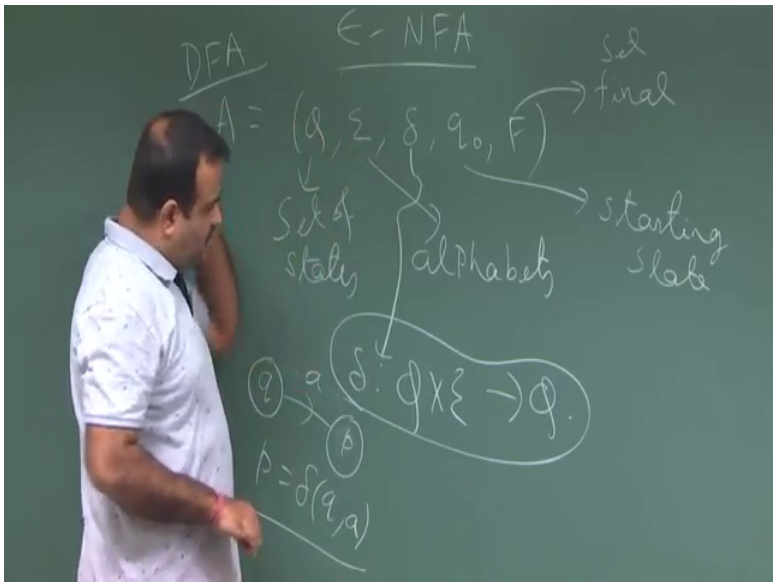
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 11
Epsilon-NFA

So, we will start the Finite Automata with Epsilon move that is this is a new concept ok. So, so far what we have seen in finite automata? We have seen DFA it is a 5 tuple.

(Refer Slide Time: 00:27)



So, this is a finite automata a 5 tuple, where this is a state set of states set of all possible finite states; finite number of states. Set of states and this is the again the finite set of alphabets input alphabet and this is the delta; delta is a function form for DFA.

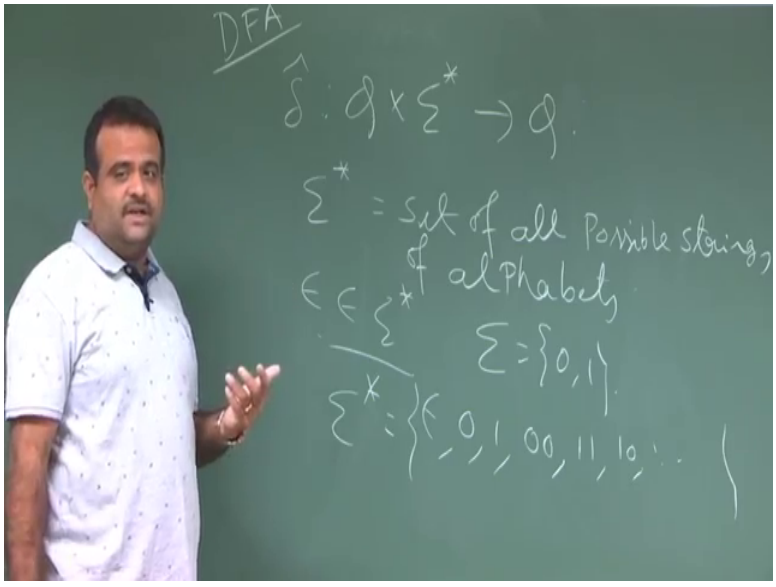
This is Deterministic Finite Automata DFA, delta is a function form Q cross sigma to Q so; that means, we are at a state q and we take an input a. So, delta of q a we take input a and it will reach to another state say p. So, we can say p is equal to delta of q comma a ok.

So, basically in you know in a finite automata we have given a string of alphabet that we put it into a in a tape and we keep on read the tape. So, once we say we start with the state q 0. So, if we had state q and if we see a input alphabet a we will go to the state p.

So, we have a function in a complete function for this for every state and every input alphabet where we are going that table we have that is the state transition table. And this is the starting state or the initial state and this one is the final state set of final state this we have seen.

And for DFA this is the transition function and for NFA I will come to NFA again. Now after that what we did? We extend this extend this delta to delta hat. So, we extend this delta to delta hat.

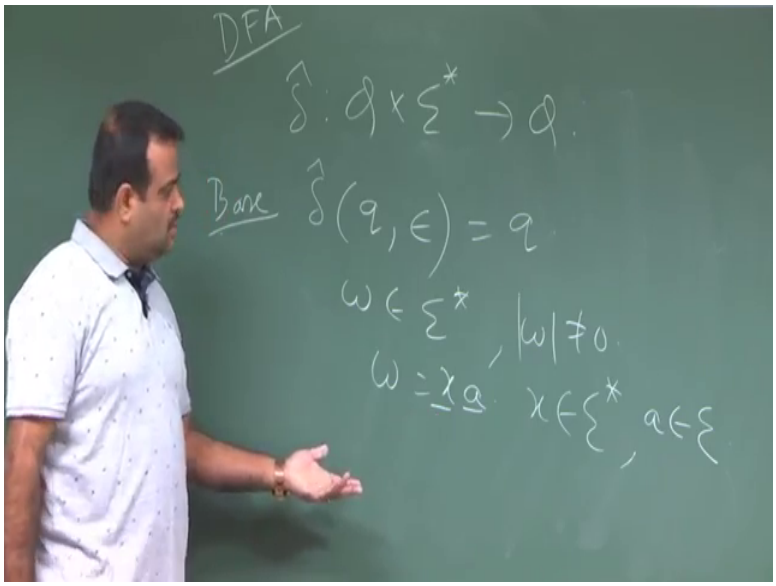
(Refer Slide Time: 03:00)



So delta hat we are still in DFA. So, now delta hat we want a function from Q to sigma star to Q. Sigma star is nothing, but set of all possible strings. So, this is the this is set of I mean the x x set of all possible string set of all possible strings of alphabets and it could be empty string also, so it could be empty string also.

So; that means, that is epsilon; epsilon which belongs to sigma star and say sigma is sigma is the set of alphabet. If we have say sigma is equal to 0, 1 binary input binary alphabet. Then the all possible combination of if epsilon 0, 1 then 00, 11, 10 this is of length 0, this is of length 1, length 1, length 2. So, all possible combination of this set is called sigma star ok. So, this is now we want to defined a we want to extend this delta to delta hat where we accept the string. So, this how we define that? We define that recursively.

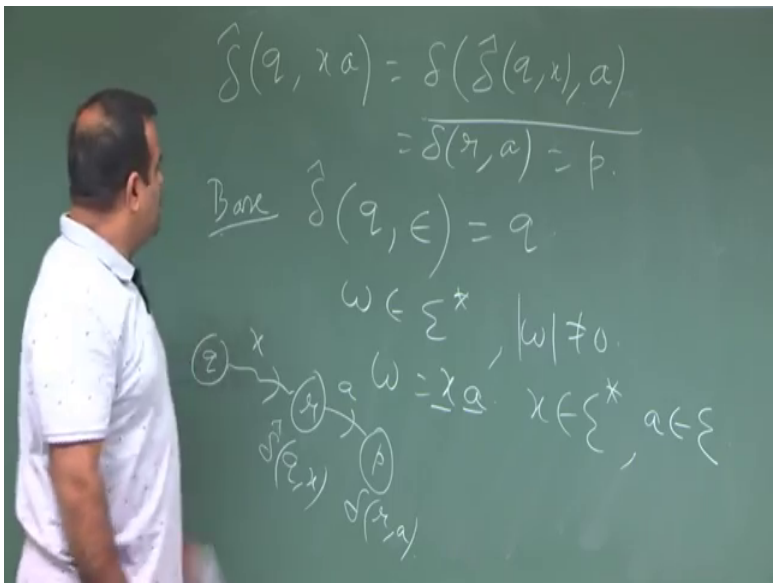
(Refer Slide Time: 04:45)



So, delta hat of if we have a q, if we take a no input like if we are not reading anything from the tape this will be remain at q. This is the base case, this is the inductive definition.

Now, if we say w is belongs to sigma star and w is not equal to length of w is not 0; that means, w will be written as some x a. So, x is a string a is a alphabet, so x belongs to this and a belongs to sigma, x is a string and x could be null also. Because this w could be a length one string ok. Then how we defined this?

(Refer Slide Time: 05:40)

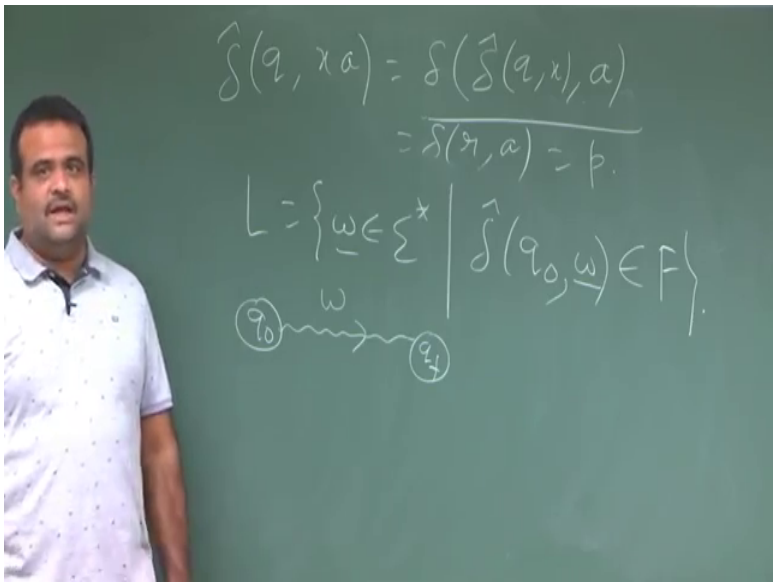


So, then we defined this delta hat of q, we start with q x a. So, we are at q now we takes we are reaching to some states say r, this is with x. So, this is basically delta hat of delta hat of q comma x.

Now after that we have a input say this is going to p. So, this is nothing, but this is basically delta of r comma a. So, this is the way we defined recursively we keep on read the tape and we reach to a state. So, this is nothing, but delta of delta hat of q comma x then a.

So, delta hat of q comma x is nothing, but r. So, this is nothing, but delta of r comma a. So, after reading the string x we are at r's r state. So, then from there we take another alphabet and we reach to the p this is the way we defined. Now in DFA we know the a string is accepted or we define that as a regular language.

(Refer Slide Time: 07:10)

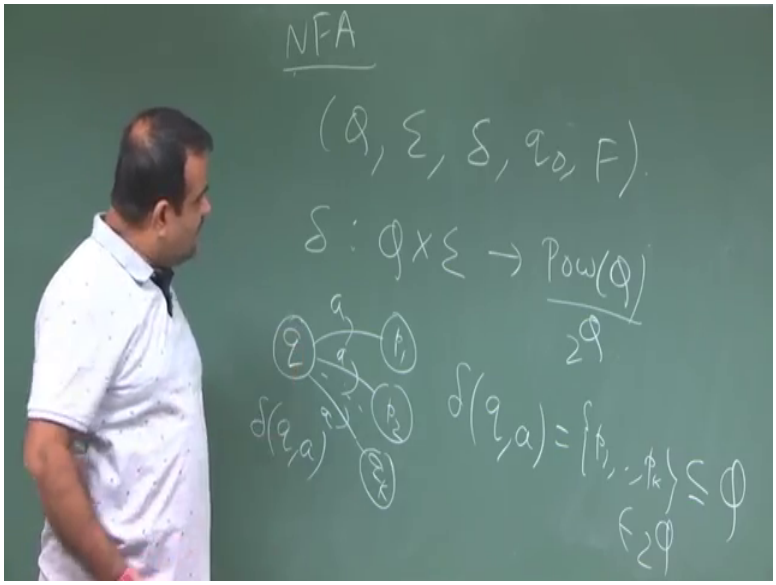


If so we define this w, this is a if we start with q 0 and if with w we reach to a final state. So, if this is belongs to final state. So, that means, we start with q 0 we have a string w, w is a string.

So, we keep on reading the string and at the end if we are reaching to a state q f which is a final state. Then we say the this na this is the language of the DFA and this is and this collection of this language is called collection of this string is called language of the

DFA. And we know that this is basically a regular language. So, any regular language you have a corresponding DFA. So, this is the DFA.

(Refer Slide Time: 08:16)



And now again NFA this is again a 5 tuple non deterministic finite automata, NFA again a 5 tuple Q sigma delta q 0 F. So, here remaining symbols are same this is the finite state of finite states a set of all possible finite states.

This is the input alphabet that set is also finite and now here there is a change in the NFA. This delta is now non deterministic; delta is now a function from Q cross sigma to power set of Q, but this is also sometimes written as 2 to the power Q.

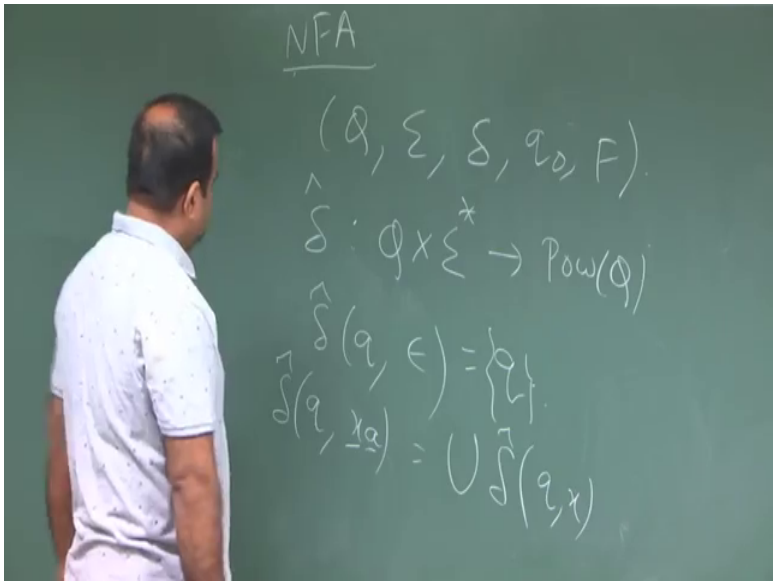
So, this is now it is going to a set in set of a set of states in set of a particular state. So, now that that is the sense it is called non deterministic. So, we have a option there like we are at q. Now if we see a so we want to see delta of q comma a.

So, now it is not a particular state; it is a set of states. So, these are all basically dot dot dot k. So, these are all now so that is why it is a this is a power set power of this is basically the set of all possible subset of q. So, delta hat of q comma a is nothing, but this set p 1, p 2, p q which is nothing, but a subset of q, which is a member of power set of q ok.

So, that since the sense it is called non deterministic, I mean it is not going to a particular state we have a option there we can go to any one of this p i's by the move off by the

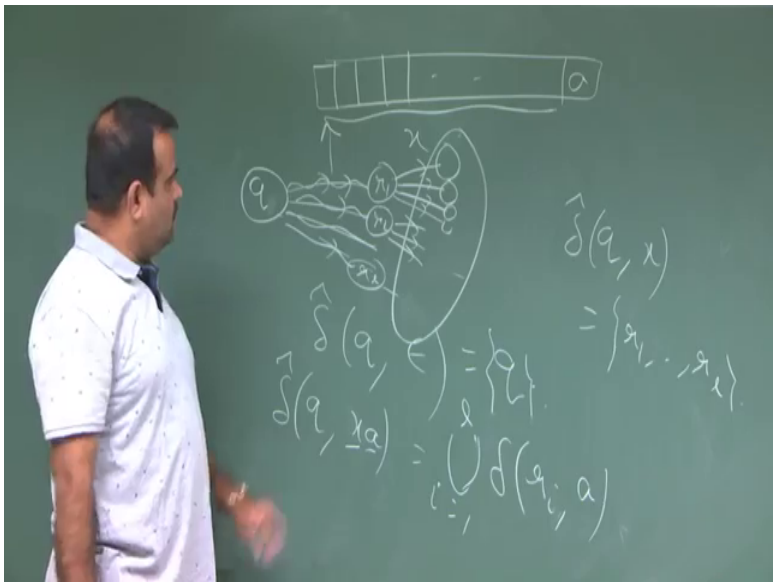
input tape a ok. So, that is the sense it is called non deterministic finite automata. And again here also we have extended the this delta hat.

(Refer Slide Time: 10:38)



So, we have extended delta hat from delta like. So, this is now so delta hat of q comma epsilon is nothing, but q, this is a this is the best case. And similarly delta hat of q comma x w; w is basically x a; x is a string is alphabet this is nothing, but union of delta hat of q comma x.

(Refer Slide Time: 11:17)

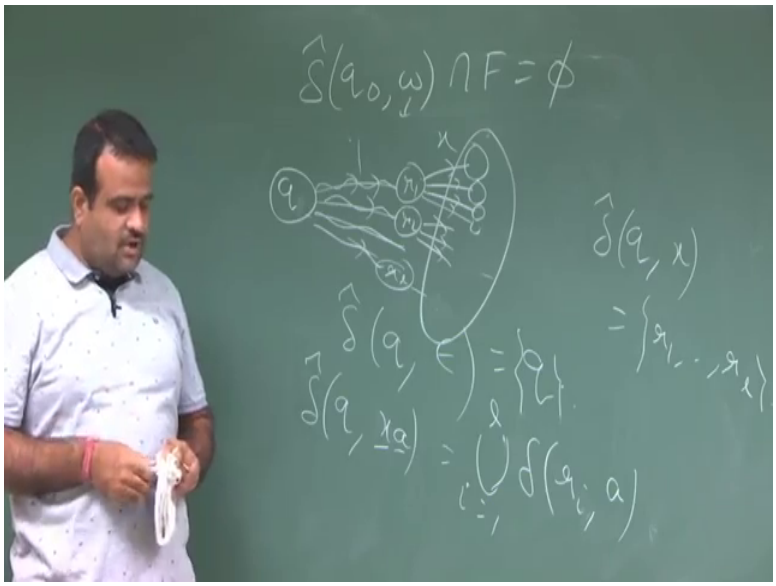


Now the q comma x so, what we are doing we have a tape over here where we have the alphabet set. So, up to this we have x and this is our a and we are reading this step; we are reading this step. And suppose we start with a set q 0 or q ok.

Now, since this is a non deterministic finite automata; so after reaching this we can go to different state say r 1, r 2 dot dot dot r rl. So, these are the possible state after seeing this. So, delta so; that means, delta hat of q comma x is the set r 1, r 2, rl.

Then from there now we have to see a. Now to see a; a that will be delta hat of this. So, that will be; that will be basically delta of r i a. So, this i is 1 to a; what is this ri's? R i is nothing, but by seeing the string we go to the these states. Now from here we can have we have the direct arc. So, these are the possible state of delta hat of w a, so this is the way

(Refer Slide Time: 13:03)



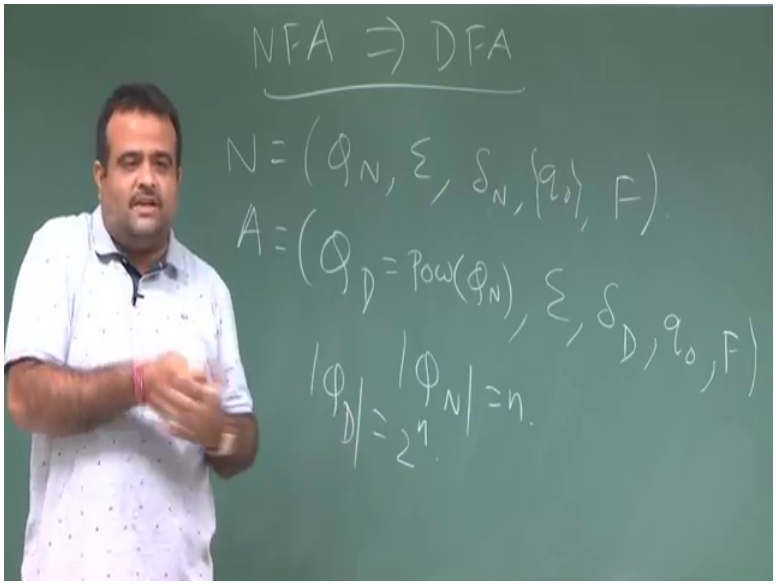
Now, if one of these branch is going to a final state then we call that w is a string accepted by this NFA. So, that means if the w have if this w intersection with F is not null. So; that means, in one of the branch we can reach to a final state because this will give us a set of states.

And in this set may not be all states are the final state we do not care that. Only thing we need at least one of the branch should reach to your final state. So, if there exists one of the branch which is reaching to a final state then it is called then this is this string is

accepted by this NFA. And the set of all such string is called the language of the NFA and this is also a regular language.

So, for a language for a regular language you have a DFA now we have also seen that NFA to DFA. Because that regular language you have seen we have the corresponding DFA. Regular language means if it is accepted by if it is a language of a finite automata.

(Refer Slide Time: 14:06)



So, again we have seen by NFA to DFA. So, what we did there? We just we have a set of states, now the DFA states are if the NFA is if this is our NFA delta N q 0 F. If this is our NFA then how we construct a DFA? DFA the Q Q D is nothing but the power set of Q N and this sigma is same and delta; delta D, delta D we have defined and this is q 0 is same F.

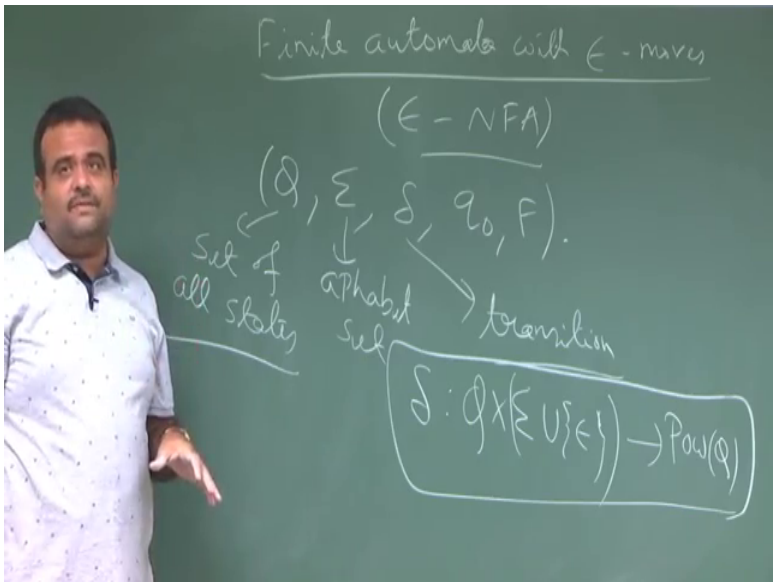
So, we have seen this how we are doing this? So, we can just we consider the set of all possible state if this has N states. Then theoretically this cardinality should be 2 to the power n, but it is not 2 to the power n because all the state is not reachable from the q 0, so we do not consider those states.

So, we just considered those states which are reachable from q 0 which contain like q 0. Because, our main objective is to find the language which is accepted by this. And we have last class we have seen one example how we can convert a NFA to DFA and they

are here taken those states I mean those say subset where it is reachable from q 0. So, they are equivalent.

So, NFA and DFA are equivalent, if you have given a NFA I can construct a DFA and DFA is already NFA. Because DFA; NFA is the superset; DFA is the subset you can say because the way the delta is ok. Now, we will introduce the epsilon move in the finite automata.

(Refer Slide Time: 16:07)



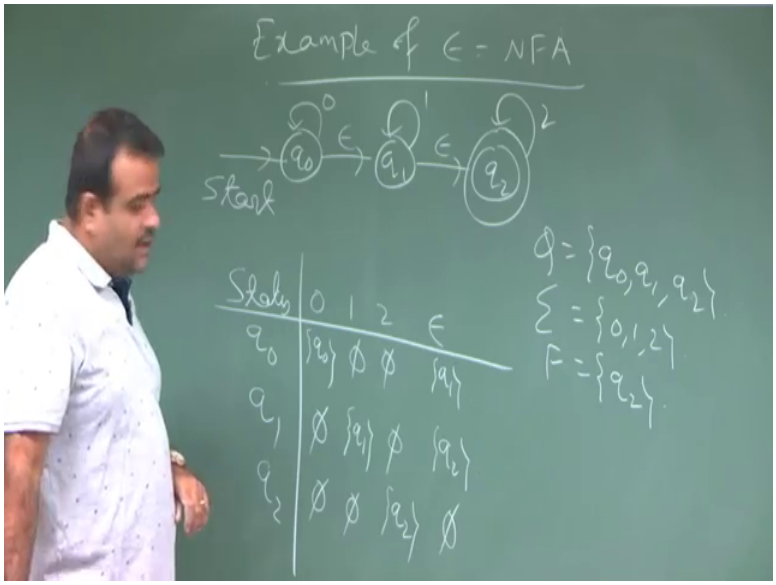
So, finite automata with epsilon moves this is called epsilon NFA ok. Epsilon move means without reading a string without reading an alphabet our machine should move to one state to another state. So, this is a extra feature we are adding to this automata. Why we need this? Because sometimes for the programming this is a added a extra feature in the programming language sense. Because sometimes we may need to move without reading a alphabet. So, that we will see later state the useful where we can use that.

But let us first define the epsilon NFA. So, again this is a 5 tuple, Q, sigma, delta, q 0, F. The Q is same set of all this is the set of all possible finite state set of all states which are finite. And this is the alphabet set which is finites alphabet set and this delta transition function you have to defined here. Because here we are allowing the epsilon move and this is the q 0 is the starting state and F is the final state. Some state we defined as a final state.

So now how we defined this delta? So, now here we are allowing the epsilon move; that means our automata can change the state without any input, so that is called epsilon move. So, without any input alphabet it can change the state. So, that is why this delta is now we are at a state. .

Now sigma union epsilon you need to add here because we are allowing the empty alphabet also a input. So, this is going to power set of Q because this is NFA. So, this is the change there from the NFA to epsilon NFA we are introducing a new input which is null basically. So, without seeing any alphabet any alphabet in that tape I mean it can change the state and this is remained same. So, we take an example so this is called epsilon move, this is a added feature.

(Refer Slide Time: 19:11)



So, we take an example of epsilon NFA ok. So, this is one example. So, you have 3 states q 2, among this is the starting state, and this is the final state. And we have the transition function like this is epsilon, 0, this is 1, this is epsilon and this is 2 ok.

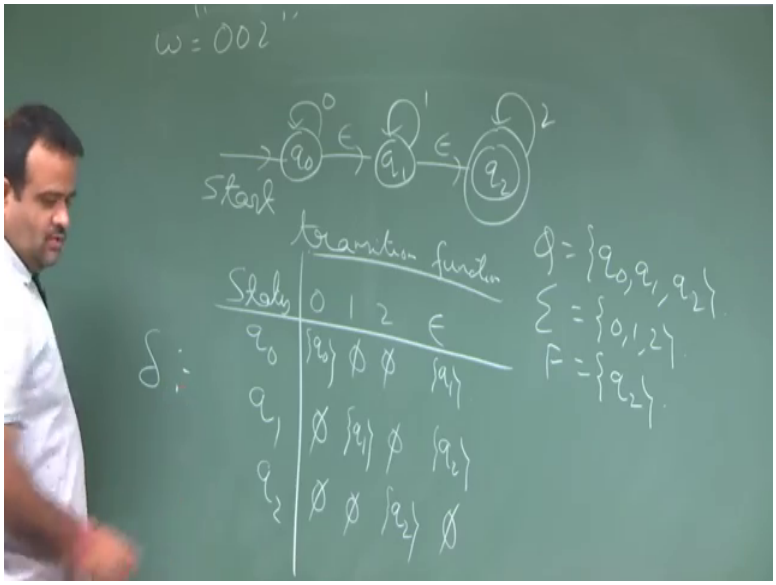
So, here our Q is 3 states; q 0, q 1, q 2 and our set of alphabet is 0, 1, 2 and q 0 is the starting state and final state is only one state in the final state this. And we have this transition function if we want to write this in the matrix. So, this is we can write as this. So, these are the states, so these are the input alphabet. So, we can have input alphabet like 0 1, 0, 1, 2 and also epsilon ok.

So, if we had q 0, if you had q 0, if we have a 0 input it is going to q 0 again and there is no input is 1 and q and we have epsilon move. So, this is empty this is empty and for epsilon we are going to q 1, epsilon means no input you have to be careful epsilon is just a symbol, it has a has no input null string. We are not reading the tape still we are moving so that is the added feature here.

So, from q 1; from q 1 there is no move for 0, this is empty and for 1 we are at remain q 1 and 2 we are at empty, 2 we have no move and epsilon we are going to q 2 and from q 2 no move for this, no move for this. We have a move this is q 2 or 2 and this is we have no move for null string in null null alphabet, null string ok. So, this is the a this is an example of epsilon NFA.

Now, we can see this is the NFA and also it can move without seeing a input, so that is why it is called epsilon NFA. So, we can just see what are the string it can accept like is it accepting this string 002?

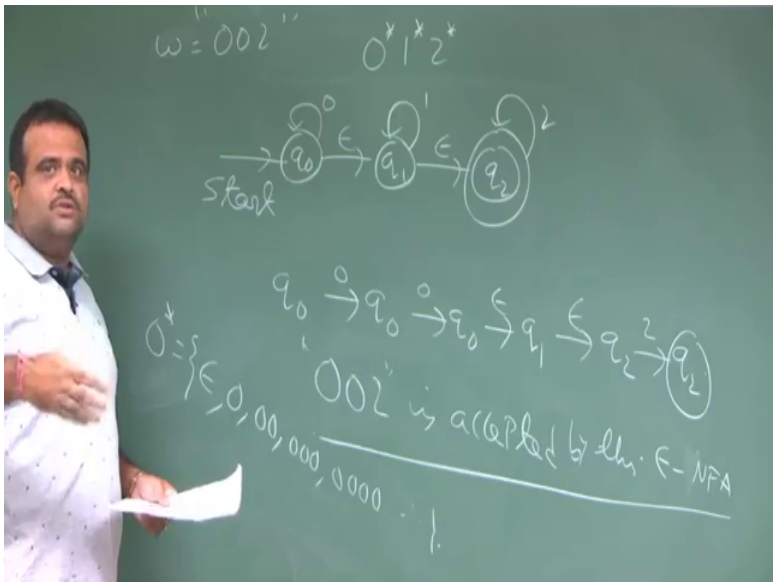
(Refer Slide Time: 22:34)



So, again in epsilon NFA we can have 002 to w, we have to define w. So, we have to extend this delta this is the transition table, transition function, transition rule. This is the transition function this is our delta.

Now, we have to extend this delta to delta hat. So, we will do that formally, but before that so we can take this 002. So, for 002 if you start with q 0. So let me rub this.

(Refer Slide Time: 23:26)



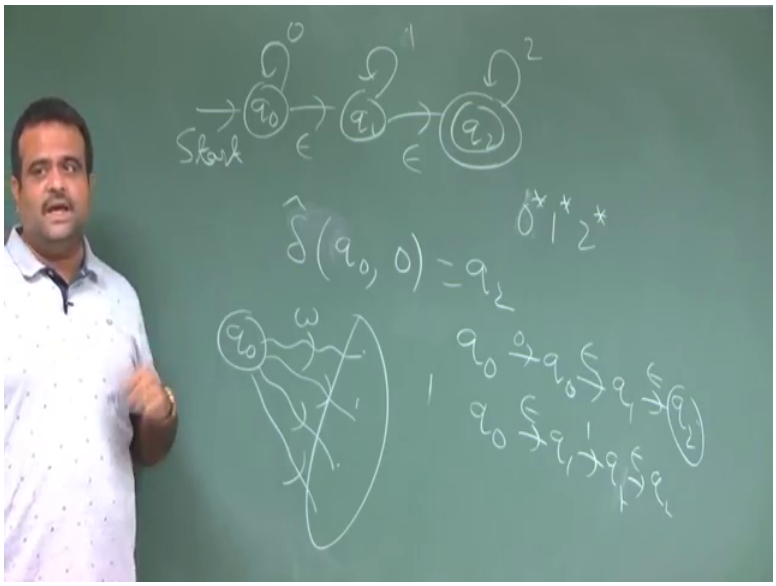
Now this so we can start with q 0 then with the 0 we can remain that q 0, then again we have 0 we can remain at q 0. And then we can have a epsilon move like without reading the alphabet we can thus the automata can move from one state to another state q 1. And again you from q 1 we can have epsilon move q 2 and then from q 2 we have a. So, you can see 2, we can reach into a final state. So, this 002 is accepted by this epsilon NFA ok.

So in fact, this is accepting all the string like 0 star; 1 star, 2 star. So, we will define the star when you talk about regular expression this is coming from a regular expression, 0 star means it is either 0 or it is 1 or it is 0011. So, 0 star is defined as it is 0, 00, 000, 0000 like this.

So, this is 0 star, 1 star is like that, 2 star is like that and this product means we will have formal definition when we talk about. So, it is all possible combination. So, this is the any number of 0's any number of 1's any number of zeroes followed by any number of 1's followed by any number of 2's.

So, this is how we defined by this symbol this symbol is called regular expression. So, we will formally define that ok. So now we will just talk about delta hat.

(Refer Slide Time: 25:49)



So, before delta hat let us just see what is that is what let us just go back to this picture again q 1, q 2. So, we have a this is the starting state this is 0, this is 0, this is sorry this is 1, this is 2 and we have epsilon move over here like this ok.

So, now we have seen this is accepting this it is accepting the all string like this, it is accepting 0. Why it is accepting 0? Because if we define delta hat, we will formally define delta hat delta hat is nothing, but we are at a state q 0 or q. So, we are reading the w, now if you read the w it will reach to many state because this is a NFA. .

Now if all of these state is final state then we call w is a accepted by this NFA. So similarly so can we say delta if we start from q 0, 0 we can reach to q 2 how? Because from q 0 we can just put a 0 we at q 0 then we see no input, q 1 we see no input, q 2 this is a final state. So, 0 is accepted even 1 is accepted because from 1 from q 0 we take epsilon move to q 1. Then q 1 we take a epsilon 1, q 2 sorry it is remain q 1 and then we have a epsilon move to q 2.

So, this is the added feature with the epsilon move it is easy to construct a automata for the given language ok. And later on we will see the this is an equivalence I mean epsilon NFA is equivalents to the NFA and then we know the NFA is equivalence to.

So, all the strings like 0 01 is also accepted. So, that is why it is basically accepting 0 star, 1 star, 2 star ok. So, we will formally define the stars this regular expression. So in

the next class we will just do the more on this we will define the extended delta hat. So, then we define the epsilon closure for that. So, we will discuss next class.

Thank you.