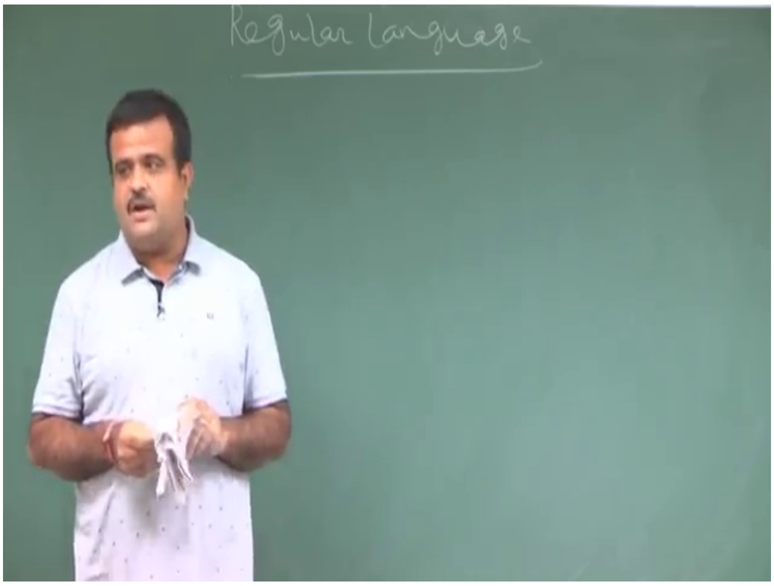
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

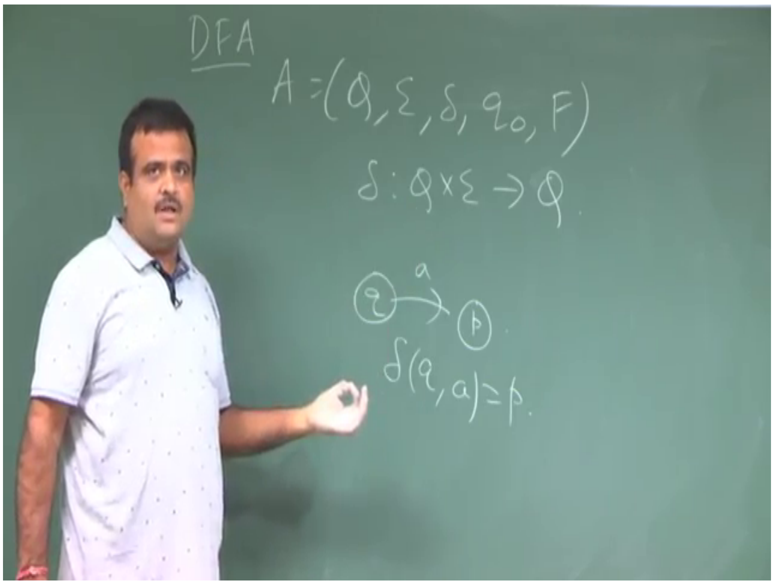
Lecture – 03
Extended Transition Function

(Refer Slide Time: 00:17)



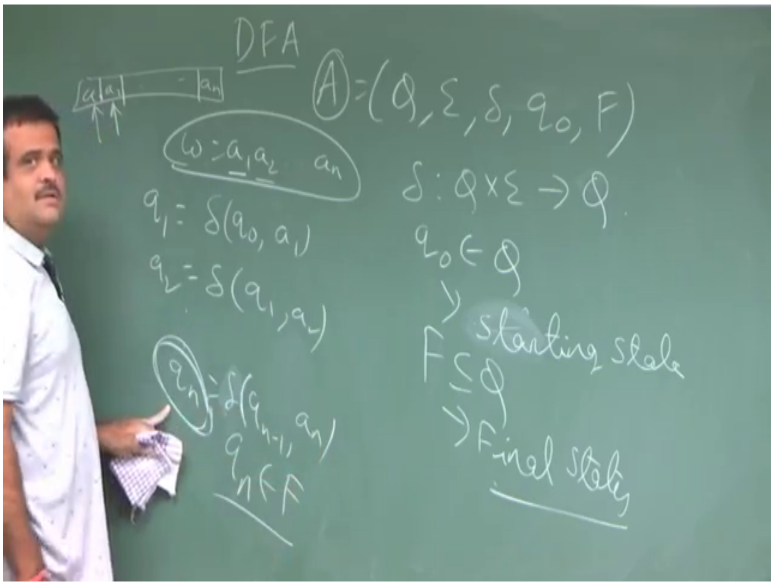
So we are talking about how a string will be accepted by a DFA. So, then we will move to the regular languages. So, the regular language is basically the set of all string which is accepting by that DFA is called by a DFA is called regular language.

(Refer Slide Time: 00:37)



So, just to recap so you have a DFA. So, DFA is 5 couple. So, this is the set of all possible state this is the set of all possible say finite set of these are all finite, is finite, number of state is also finite. This is the set of input symbol, this is the delta; delta is the function. This is the rule I mean at current state say q and we take a input say a will go to p so, delta of a, q comma this. So, we take so, this is a function from Q cross this is the state and take a input alphabet it will go to the new state and then this q 0 is a starting state.

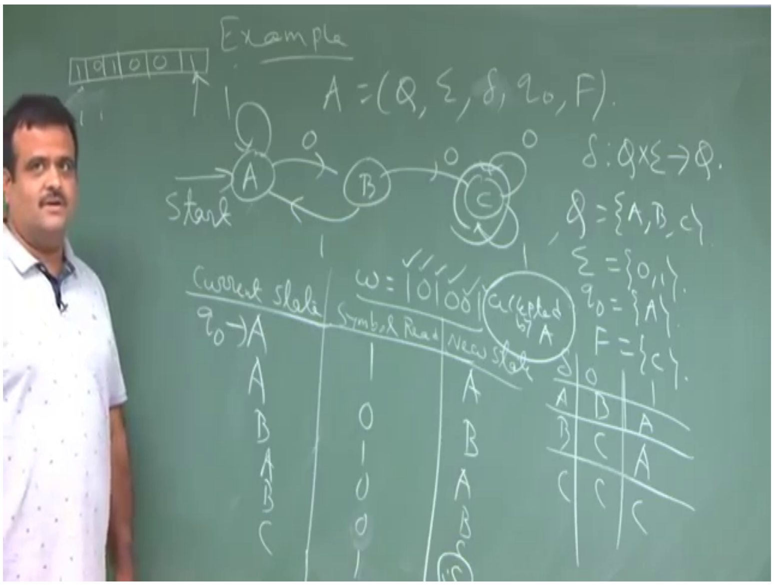
(Refer Slide Time: 01:29)



So, q 0 is this is the initial state or call starting state, starting state. And, then the F; F is a subset of Q which is call final state or the accepted state, ok. So, now this delta is defined on a on a single input, now we have seen how we can read a string; I mean the collection of input sequence of input by a DFA. So, this is like this; suppose you have a, a n.

So, we will start from the we will start from so, we will start from the starting state will first read this; supposedly reaching to q 1 and then again we are at q 1, we again read a 2. Suppose it is q 2 like this. So, finally, delta of q n minus 1 and we read this. So, this is the so, we have this a 1, a 2, a n we are keep on reading this and every time we are changing our state and finally, if we are reaching to a q n. And if q n is this is the way we read a string instead of if q n is belongs to f final state then we say the string is accepted by our DFA otherwise the string is rejected, ok.

(Refer Slide Time: 03:29)



So, this is the way we define. So, we take an example we take an example consider that the DFA like this. So, we have state A, B, C B and C. So, A is the starting state set, A is our t 0. So, from A if we see a 0 will go to B and if we see a 1, we will remain at A and from B so, if we see a 1 and come here and if we see a 0 will go there. And from C if we see so, this is the and this we define as a final state. So, this is sigma, delta, q 0, F.

So, what is our delta over here? So, delta is nothing, but so, Q is A, B, C and sigma is binary string binary symbol 0, 1 and our q 0 is the A which is the starting state and F is C the final state and delta is the rule. So, we can have a delta this. So, we have A, B, C 0, 1.

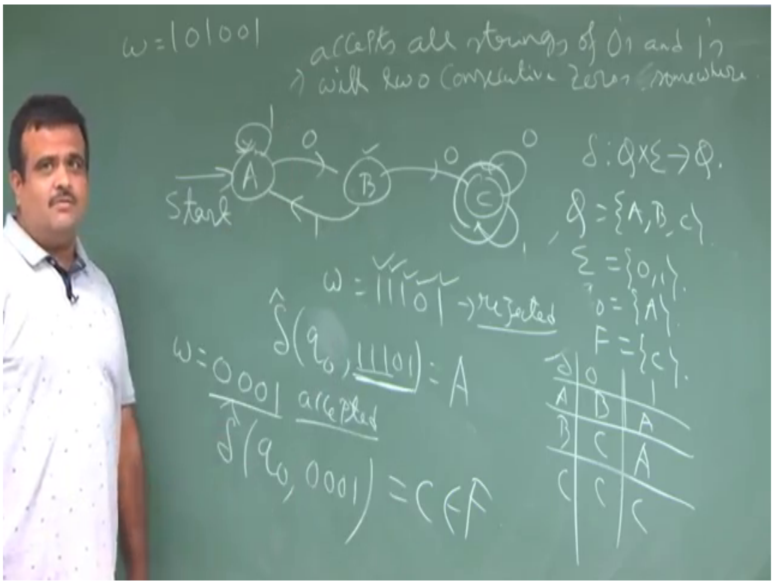
So, for from A we can go to 0, we see B and from B if we see 0 we go C otherwise we go back the. So, this is the this is a DFA and that is the rule, transition rule; that is the transition function which is a function form Q cross sigma 2 Q. So, we are at some state we see the input depending on that rule we will go to the another state, ok.

Now, we will see how we can read a string by this DFA. So, suppose you have a string say 1 0 1 0 0 1; this we want to take the input and we execute this on this. So, we start with so, this is our current state and this is our say symbol, symbol we are reading and this is our new state ok. So, we start with the A that is the q 0 starting state. Now, we are reading the symbol 1. So, now, if you at A we read 1 will remain at A this is the new state, ok. Now, again so we are at A we are seeing 0 then so, if you see 0 we will go to B. So, we are at B. So, this is done, now we are seeing 1. So, in B if we see 1 will come back to A. Now we are at A again, will see 0; if we see 0 will be again at B. Now, we are at B, now if we see 0 again we will go to C.

Now, C is a final state, but our string is not yet end. We cannot say that in the middle we are reaching to the final state, so, will not execute the remaining part of the string, no. No, that that is not allowed we have to explore whole string not the truncated string, ok. So, we are at C now. So, yeah we are at sorry we read 0, we reach to the C. Now we are at C, now we read 1. So, we reach to C again that it by that rule, ok. So, basically we put these in a tape we can say this 1 0 1 0 0 1 and we read this tape. So, this sequentially and depending on the rule delta our state will be changing. And, at the end of this if we reach to a final state, this is happened to be a final step. So, that is why this string is accepted by this DFA.

So, this w is accepted by A because at the end of the string it is reaching to a final state, ok.

(Refer Slide Time: 08:55)



Now, we can take another string say. Yeah I should take w to be say 1 1 1 0 1, ok. So, we start with here, we see a 1, then maybe we remain here. We see another one, so, this a new state we remain here, we see another one with this the new state we will remain here we see a 0 once you see a 0 will go here. Then we see a 1, if we see a 1 we will come back to here. So, we are reaching to A after execution of this.

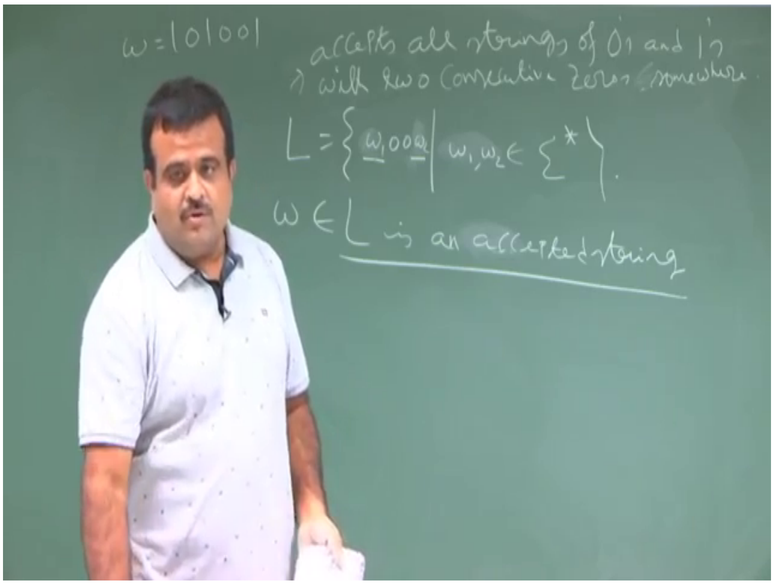
So, this is we can define a delta hat. So, delta hat because delta is defined on a alphabet input and delta hat we can extend it, we will define that. We can extend it over a string. So, delta hat off this 1 1 1 0 is nothing, but A. So, the meaning of these after execution of this whole string where we are at we starting with we are starting with q 0 that is the starting state, we keep on reading this string. I we read this string, so, at the end of the string where we are. So, this is our A; A is the distant. So, this is same way this is the DFA. So, now, A is not a final state.

So, this string is not rejected. It is not a accepted string, rejected by our DFA, ok. Now, if you take say 0 0 0 1, this is our w. Now, what is delta hat off? You start with q, 0 0 0 1 you have to see it. We are not telling delta hat because delta is defined on the single input alphabet and delta hat is we are extended it to over a string. So, we read this string after that it is giving us which state, ok. We start with q 0 that is our A, so, we read 0 it will go to B we need 0. Again it will go to C, then we read 0, remain C, we read 1 remain C. So, this is our C which is say accepted state. So, this is accepted. This string is accepted by this DFA, ok. So, this is not accepted.

So, what was the earlier string? Earlier string was you can have a we want to write this set of all possible string which is accepted by this. So, whether we can write it or not. So, 1 0 1 0 0 1, this was accepted then this is accepted, this is not accepted. Any other string? So, can you have a general formula for accepting this DFA? Can you write this? So, is it accepting the accepts all strings which has two consecutive 0 because, to accept we have to reach to C. So, how we can reach to C if we can see two 0's, two consecutive 0 because we are at A. To reach to C, we need to add two consecutive 0, that is all.

So, it is accept all strings of 0's and 1's with two consecutive 0 with two consecutive 0's somewhere, we do not care where, somewhere; like this one. You have many 1's, this is not accepted, but if we have a another 0 over here this could be accepted. So, this is the so, it is accepting all the string which is having two consecutive 0 somewhere, ok. So, can you write this in a like general way. So, can we just write these as this.

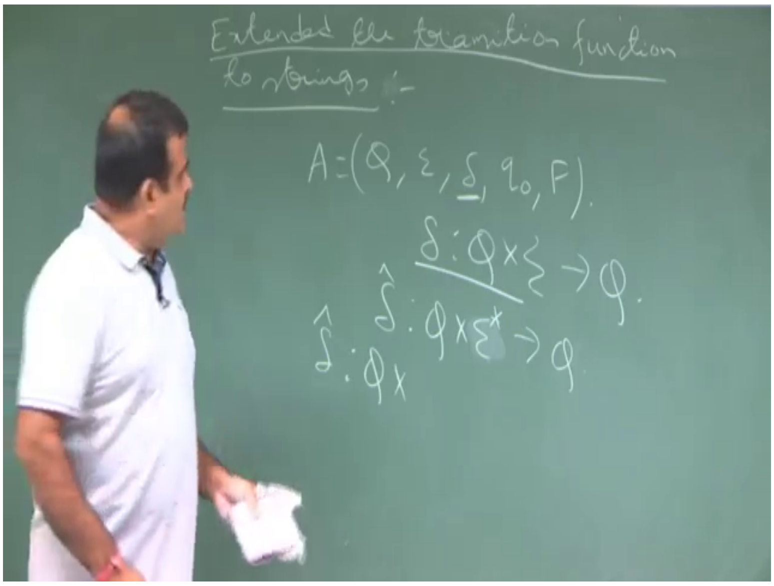
(Refer Slide Time: 13:49)



So, this is again a language. So, we can just write this as a x 0 0 y kind of thing sum. So, x is x and y coming from sigma star any length. Yeah, we need to have two consecutive 0 in the middle. So, instead of x we can have yeah. So, we can just x is usually referred as input so, w 1, w 2 w 1, w 2. It could be if we have two just. So, this is if w 1 is epsilon, w 2 is epsilon this is 0 0 0 0 is accepted because we start from A, go to B, go to C. Now, if this is say does not matter if we reach to 0 0 if you take 0 0 then any string, you did not keep on hop there because in the C we are in the same state for any input; if it is 0 also C 1 also C. Now, if it is 1 0 1 or 1 1 so, this is the way we can write this. So, this is the so, any string belongs to w any string belongs to this is a accepted it is an accepted string; any string from this is an accepting accepted string, ok.

So, we can will always try to write this in terms of this I mean that maybe it is not always possible to write in terms of this in terms of a unified way say to it ok. So, now, we will formally define how we can extend this DFA I mean transition DFA transition rule over a string. We have seen the transition rule is defined on a input alphabet, but just now we have seen we can take the input as a string instead of an alphabet in the transition rule. Another input is a state so, that we will define. We will we are going to extend this delta to delta hat ok. So, this is basically extended the transition function.

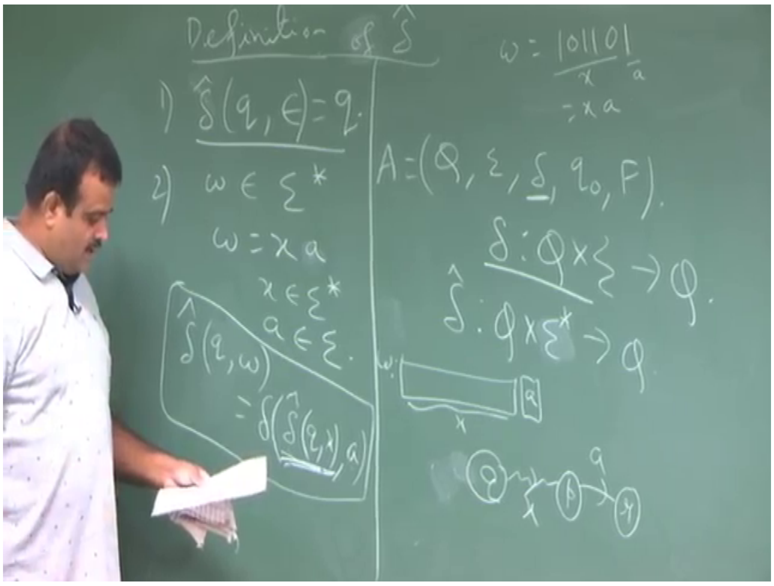
(Refer Slide Time: 16:27)



Extended the transition function over to string, ok. This is the formally we had we have already seen because in the last xo we have a DFA Q, q 0, F. Now, this is a rule which is taking to the. Now, this delta we are going to extend this over a string. So, this will take a string w and it will go to a state, ok. So, this delta is only take one alphabet, but this is checking string of alphabet; it is the formally we are going we have seen the one through one example. Now the formally we are going to define this. So, this is said then if we want to write this in this form delta hat is a function from Q cross. So, this is one input.

So, this is sigma star because it is accepting it is taking a string as a another input then it is reaching to a another input after end of the string. So, let us define this how we define this delta hat on this, ok.

(Refer Slide Time: 18:19)



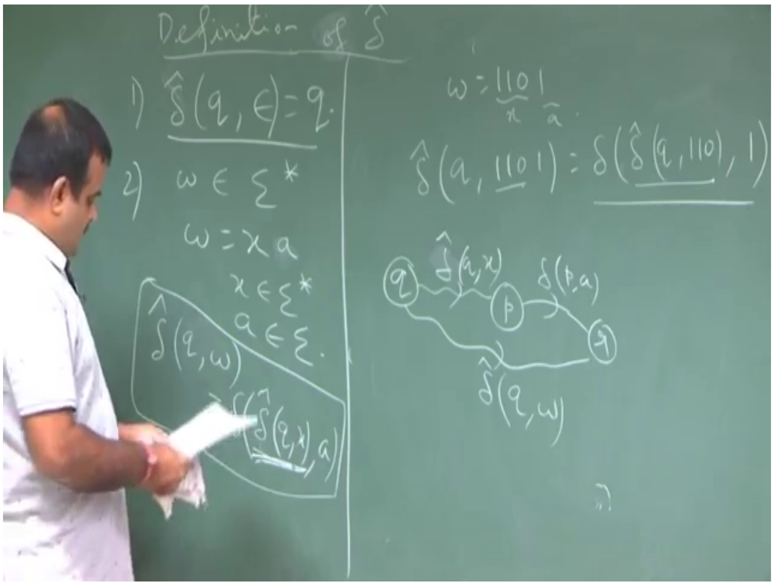
So, this is the definition of delta hat, ok. So, first of all this is taking input as sigma star. So, sigma star belongs to null set also. So, that is defined first this is the base case, if we had q we take sigma the epsilon will be at remain q, this is how we define. This is the convention, we are because in this DFA we are not having a epsilon move, but epsilon move will be there in NFA or epsilon benefit, but here we do not have any epsilon move. So, suppose we are at state q and we see epsilon; that means, we do not see any input then we will be at remain this state. So, that is the idea. So, this means no input, no input. We are not reading the tape like this. This is the base case we can say, ok.

And, then the inductive step we have. So, suppose we take w from this and suppose w is equal to sum x into x a x a and x is coming from sigma star and a is sigma, a is a input. Say for example, if w is a 0 1 1 0 1. So, this is x, this is a. So, this we write x a kind of thing, ok, the last symbol you are just keeping like this. So now, how you define delta star on this? So, delta star on so, we suppose you are starting from a state q on w.

So, this is defined as delta of delta star of q, x and then a if this is the; that means, so, this is our x, this is our w this is our x and we have a last symbol which is a. Now, we have a rule for a; now this is the inductive step. Now we start from q I mean q 0 you start, but we do not know whether we are starting from the starting state or this may be in the middle I mean. So, now, we first read this x. So, this will keep on execute and some after sometime it will reach to your state set p, then on p we can apply this delta on a. So, it will be a sum r. So, that is the rule, that is the way we are defined inductively, ok. So, we are just defined this one.

So, this is basically the sequence x; x is again a sequence. Now, if it is only a then x is epsilon. So, this will be remain in q now we take a will go to the another state. So, that is straight away the application of delta, but if it is not a if it is some more symbol then we first take this one then we apply this. So, lastly we will move to the state r. So, this is the way we define this delta, ok.

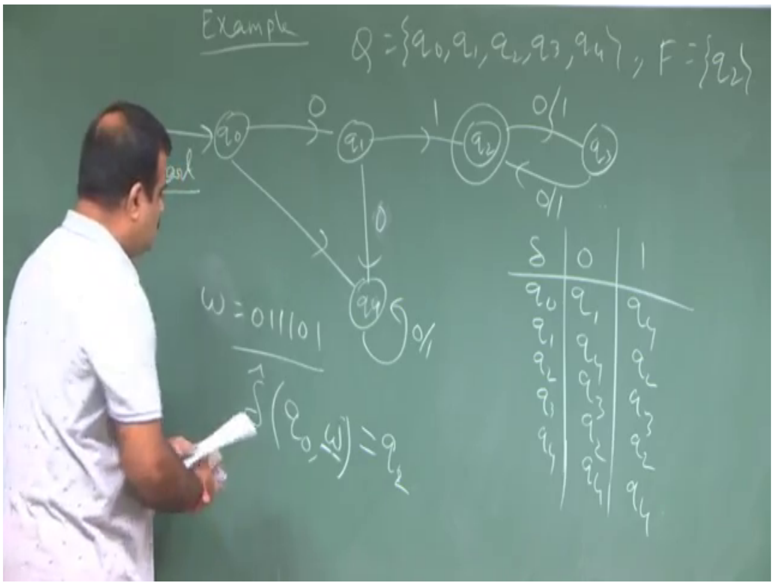
(Refer Slide Time: 22:31)



So, we take one example, say w is a 1 1 0 1. Now, we take this as a x, this as a a ok. Now, delta hat off say we are at q we are taking 1 1 0 1 as this. So, this will be defined as delta off we first read x. So, after reading x we will be at some state and that will be by this rule and then we will see a 1. So, where we are going so, this is the way we will see it.

So, we will just I mean intuitively we start with the some state q and we have we keep on so, we will go to p this. So, this is the delta hat; delta hat of q comma x will raise to p. Now from here we have direct delta so, delta of p comma a. So, this is suppose r. So, this together is our delta hat of q comma w, this is the way we are defining that. This is the inductive definition of this. So, we will take an example then it will be more clear.

(Refer Slide Time: 24:11)

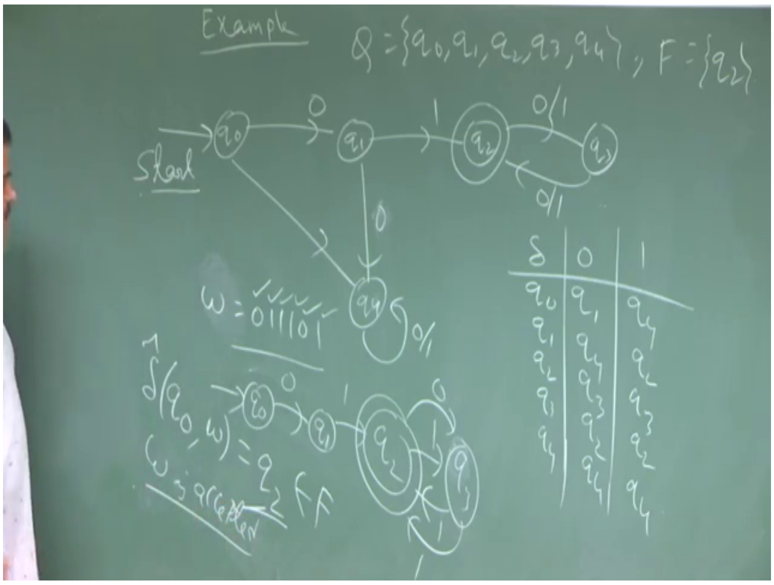


We will take an example ok. Suppose we have this q 0, q 1, q 2, q 3 little bigger and q 4 and this is our starting state and say if we say 0 over here we will go there and if we say 1 over here we will come here. And, from here if we see 1 over here, so this is our delta and from here if we see 0 over here we will go here sorry 0 confirm. So, this is our final state. So, if you see 0 or 1 anything will go to q 3 and from q 3 if you see 0 or 1 anything we will go to q 1. So, this is our and from here it which is 0 or 1 will hop there. So, this is one example of a DFA, Deterministic Finite Automata.

So, what is our Q? Q is consists of q 0, q 1, q 2, q 3 and q 4 there are 5 state, and we have this is our starting state. And what is our F? F is our q 2, ok. Now this is our rule, delta. So, delta is the rule delta come so, q 0, q 1, q 2, q 3, q 4; there are 4 state and here 0 and 1. So, if we see from q 0 if you see a 0 then we will go to q 1. And, if you see a 1 we will go to q 4 and from q 1 if we see a 0 we will go to q 4 and if you see a 1 you will go to q 2. And from q 2; from q 2 if you see 0 or 1 both are we are going to q 3 and from q 3 if you see 0 1 both are q 2 and from q 4 if you see 0 or 1 both are. So, this is our transition rule, the delta. This is not delta hat, this is the delta.

Now we want to see we want to read a string like this 0 1 1 1 0 1, ok. So, you want to see whether this is accepted by this by this DFA or not to accept it. So, we have to so, start with this q 0 and this should be the accepting state, this is a string so, that is why we put a hat. If it is just input we will use the delta. So, this should be q 2. If this is q 2, then this w is accepted. So, we will see that whether ok. So, we are at q 0, so, we read this 0 will go to q 2.

(Refer Slide Time: 27:45)



So, we are so, we are at q 0 we will see 0 will go to q 2. So, this is the starting state and from q 2 we read a 1. So, then we go to I sorry this is q 1 and we read a 1 we go to q 2; q 2 is a final state, but we have not even not yet completed the string, we have to complete it. Then we see a 1 if you see a 1 we go to we go to q 3 this final state we go to q 3, we see a 1. Again, we see a 1 if we see a 1 we come back to q 2, we see a 0 if we see a 0 again we will go to. So, we are seeing 1, again we see a 0 we will go to q 3 and they see a 1 again we go to q 2.

So, delta hat off q 0 and this w is giving us q 2 which is accepted state which is belongs to F. So, that means, w is accepted, w is accepted, ok. This is the one example. So, we can try to have a general rule for this how this is accepting. So, then in the next class we will discuss more on this and then formally define what is a regular language? Regular language, it is a language which is accepted by a finite automata. So, we will talk about this in the next class.

Thank you.