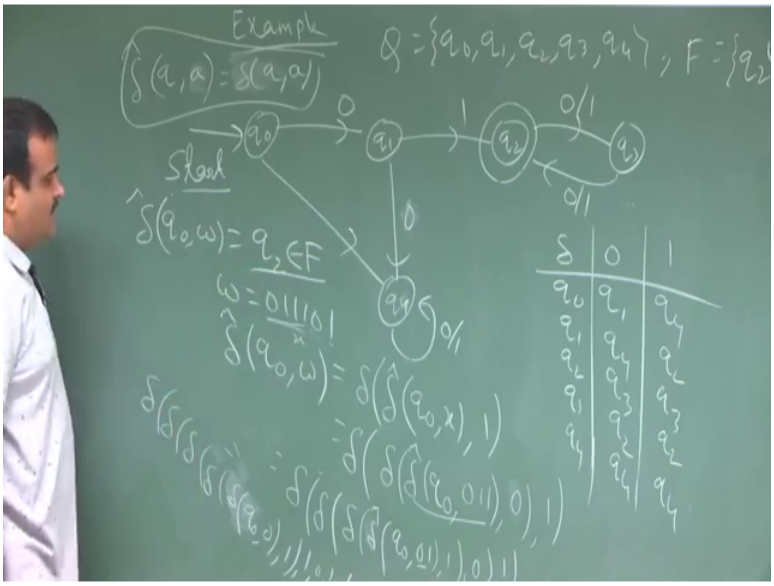
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 04
Language of DFA

(Refer Slide Time: 00:17)



So, we are talking about a language accepted by a DFA that string, string accepted by the DFA. So, this was the examples we have taken in the last class. So, suppose we have this DFA which is having the states, q 0, q 1, q 2, q 3, q 4. And the q 2 is the final state, and q 0 is the starting state, and we have this rule. And then they have seen, this language is this DFA is accepting some string like, this example we have taken w say 0 1 1 1 0 1.

So, we want to define the delta hat, we start with the q and on this w. So, how we do that? So, this is by definition of the delta hat, you recursively I mean the intact induction I mean anyway. So, this is nothing but delta hat of so these we write in x a. So, delta of delta hat of q 0 x, then we see a 1.

So, this is also delta of delta hat of this. So, x is also it can be written as this, so delta so this is delta hat of q 0; no delta of, so delta of delta hat of q 0 comma, now this is our x over 0 1 1, and then we have 1. Sorry that this one is 0, and then we have 1.

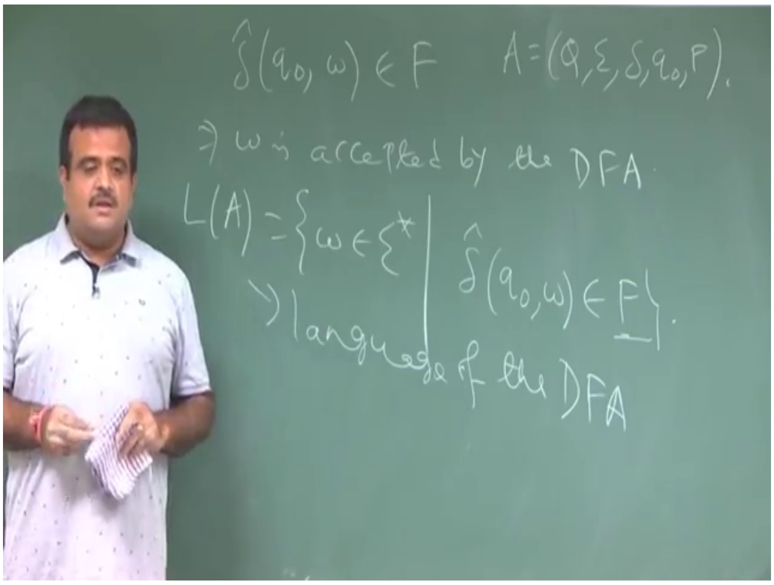
So, this way if you continue, so this is again delta of delta of then this delta hat again we can define it as delta of delta of delta hat of q 0 0 1 0 1 comma 1, then comma 0, then comma 1. So, this we can define as delta of delta of delta of, now this delta hat now we have only two.

So, this we can just define a delta hat of q 0 x, this is the delta of delta hat of you know q 0. Sorry this is delta hat, but delta hat is one symbol is basically delta, delta of 0. Actually, delta hat then delta hat of q comma epsilon. So, this is say only one symbol a, so this we can write delta hat of delta of q comma epsilon is epsilon; so this is nothing but delta of q comma a.

So, once you switch to a single symbol, it is this. So, this is delta of q, q comma q 0 comma 0, then comma 1, then comma 1, then comma 0, then comma 1. So, if you do that, so now what is the q 0 comma 0, q 0 comma will go here; then 1 will go here, then 1 will go here, then again 1 will come back here, then again 0 will go there, again 1 will come back here; so that means this delta hat of q 0 comma this w is nothing but q 2, which is the accepted state. So, this is the this is a string which is accepting, this is just a example.

So, now we define the language of a deterministic finite automata. So, such a string which is giving which is reaching us to the accepted state, such a string if we can put it into a set back that back is called language and that language is called the language of the DFA.

(Refer Slide Time: 04:49)

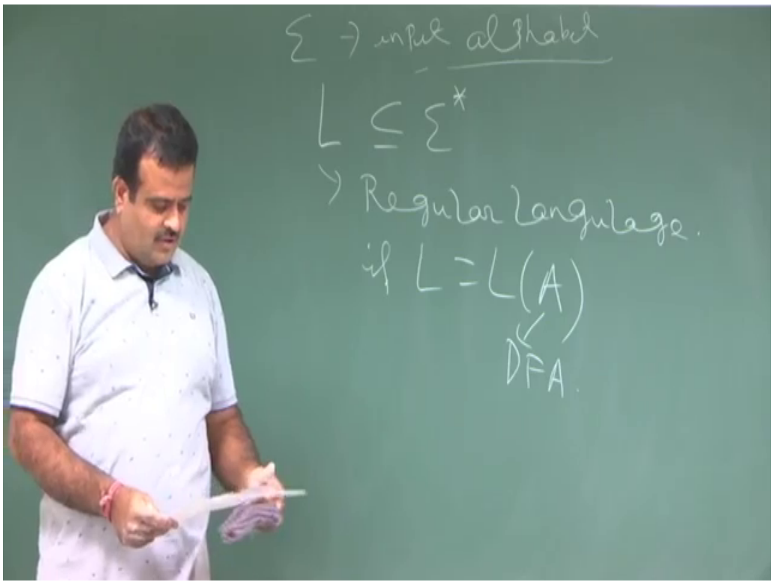


So, delta hat of q 0 comma w if it is belongs to F- Final state. So, then we say w is accepted by the DFA ok. So, this is the set such set is called the language of a DFA. So, if we this is the set of all string, such that delta hat of q 0 w is belongs to F. Then this is called language of the DFA.

So, if we if our string, so this is consists of all possible such string which is accepting which is reaching to us the accepted state. We start with the starting state, we keep on reading the input symbol and then we keep on changing the state depending on the input. And then finally, if we can reach to a final state, then that that string is accepted. Now, this language is the collection of such accepted string is called the language of the DFA.

And this is defined as, so if our DFA is A, this is consists of 5 tuple; Q, sigma, delta, q 0, F. This is the F, then this is called the L of A language accepted by this DFA. Ok, this is the set of all string w that start with the starting state and if it is end with the final state, then it is called it is called the language the set of all such string. Then the string is accepted by the DFA and the set of all such string is called the language. If we put into the back that back is the language of the DFA.

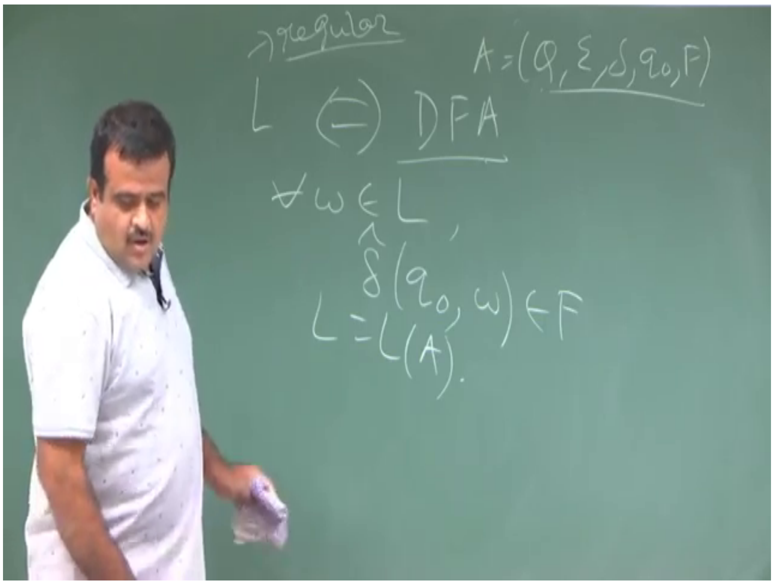
(Refer Slide Time: 07:25)



Now, we define what do you mean by a regular language. So, this is the input alphabet or alphabet symbol, input symbols. And we know the language, so language is any language is a subset of sigma star. It is a collection of string, any collection of string is called a language; including the null string ok. Then when we say this is a regular language, if it is a language of a DFA, if there is a DFA which is accepting all the string of it; then we call this is a regular language.

So, it is a regular language, if this L is L of A where A is a DFA, so that means, if the whole string if there exists a DFA which is accepting the all string, all string, all the string of this language of the string of the set then we call this is this language is a regular language. So, any such collection of this of the string which is accepted by a DFA is called a regular language. We will take some example of some language and we will determine, whether this is a regular or not. To be a regular, we need a finite automata which is accepting that all member of that set ok.

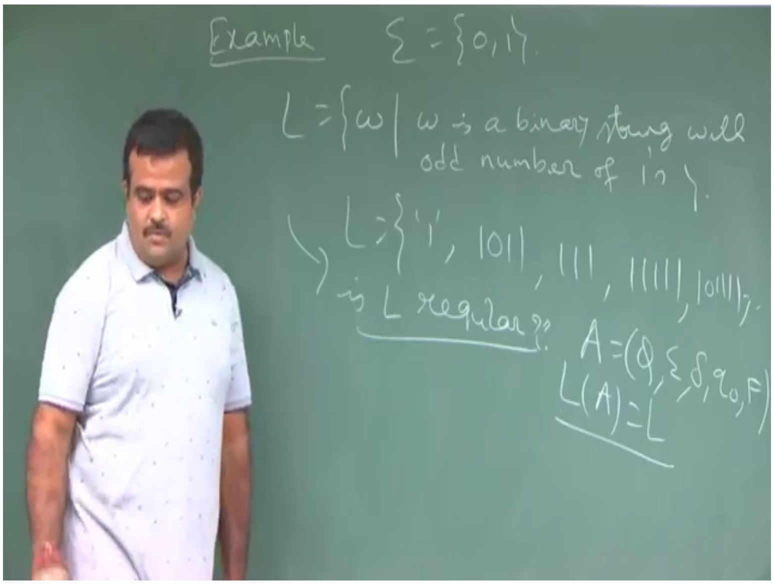
(Refer Slide Time: 09:05)



So, we will take some example. Regular language means a language will be regular, if there is a DFA, deterministic finite automata such that which is accepting this language; such that if we take any string from this, for all string from this, and if this is say q 0, F. Delta hat of q 0, w will be F, for all for all this.

Then we know this is this is called language of that DFA, then next to say this is a regular language; if is if a DFA exist, if finite automata exist, then we say it is a regular language ok. So, we will take some language and we will determine whether it is a regular or not that means, we have to find out whether there is a automata which can accept that ok.

(Refer Slide Time: 10:23)

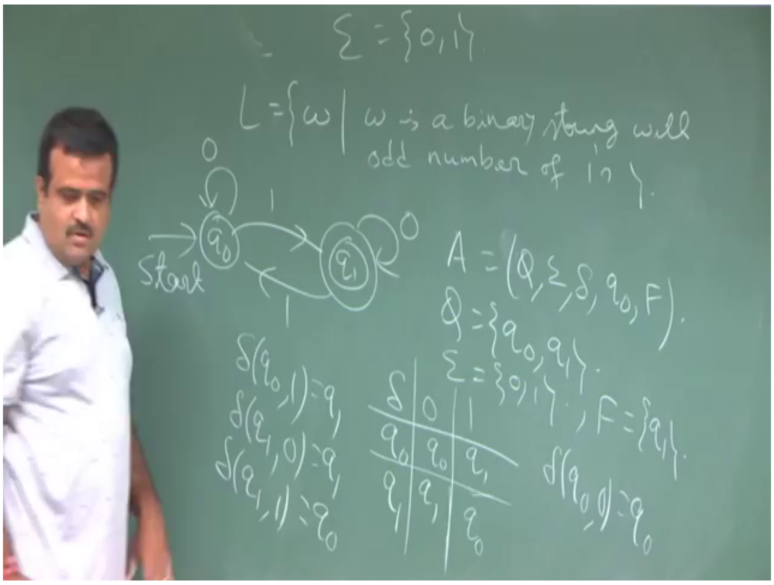


Let us take some language, some set of string. Ok, here, here, here our input symbol are binary 0 or 1. Now, let us define some language w, which is w is a string with odd number of 1's. w is a binary string, all binary string collection of all binary string with odd number of 1's, odd number of 1's that means, we have given the string, the string is coming from the input symbol 0, 1. Say it is a sequence of 0, 1 bits; 0, 1 symbols.

Now, we consider the whole such string which is having odd number of 0, like this is; this is a string, this is only 1 1. So, this is a one string, 1 0 1 1 this is another string, these are all string which is having odd number of 1's. 1 1 1 1 1 five 1's; 1 0 1 1 1 1 five 1's, so any such possible; so, this is a collection of all such stream. So, this is our language and we want to check whether this is a regular language or not. So, further that what we need to do, we need to have a to be a regular we need to have a DFA, which is accepting this which whose language is the same as this language.

So, we need to have a DFA which is accepting all the string or where the number of 1's is odd. So, let us try to so the question is this regular, regular. So, for this we need to be regular we need to have a DFA, delta q 0, F such that the language of the DFA should be same as this, did we have to check. So, let us try to find a DFA, yeah so this ok.

(Refer Slide Time: 13:13)

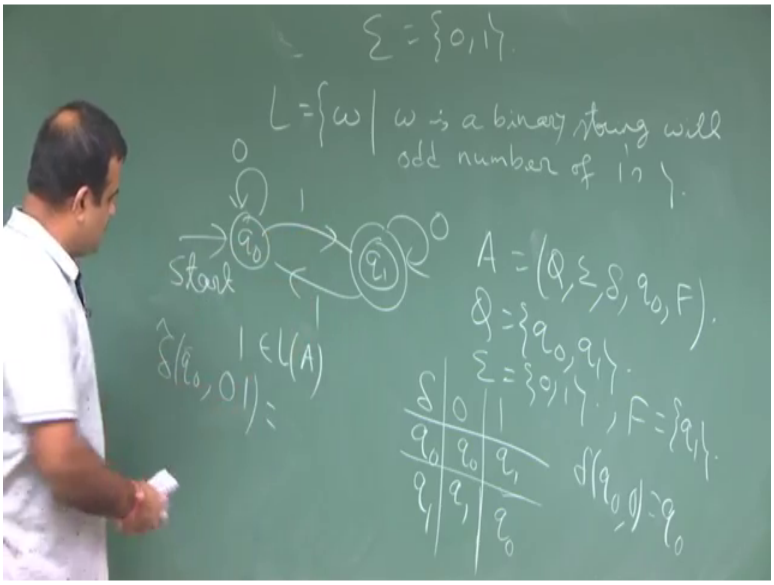


So, so let us start with q 0 is the starting state and we have a q 1. Suppose from here, if we take a input 1 we are going here, if you take a input 0. And from q 1, if we take 0 will be remain at this. And if you take a 1, you have to come back here ok, very simple DFA. So, what is the DFA? So, this is our DFA A which is consists of five tougher Q, sigma, delta, q 0, F.

Now what is Q? Q is basically only two state we have q 0, q 1 and sigma is 0, 1; you have either in finite we have binary input 0, 1. And q 0 is this one, q F is F is q 1 final state, we have only one final state; and delta delta consists of delta is the rule. So, this is from we have two states q 0 and q 1. So, if we had q 0, this rule is telling if we put a 0 input. So, delta of delta of q 0 of 0 it is q 0, because this is q 0 and delta of q 0 of 1. If you, if we add q 0 state, if you read a 1, then we will go to q 1 that is the rule q 1 ok.

Now, from q q 0 delta of x sorry q 1 if we put a see a 0 input will be remain at q 1. And from q 1, if you see a 1, we will come back to the q 0 that is the rule. So, this is a finite automata. Now, we have to check what are the type of language the automata is accepting ok.

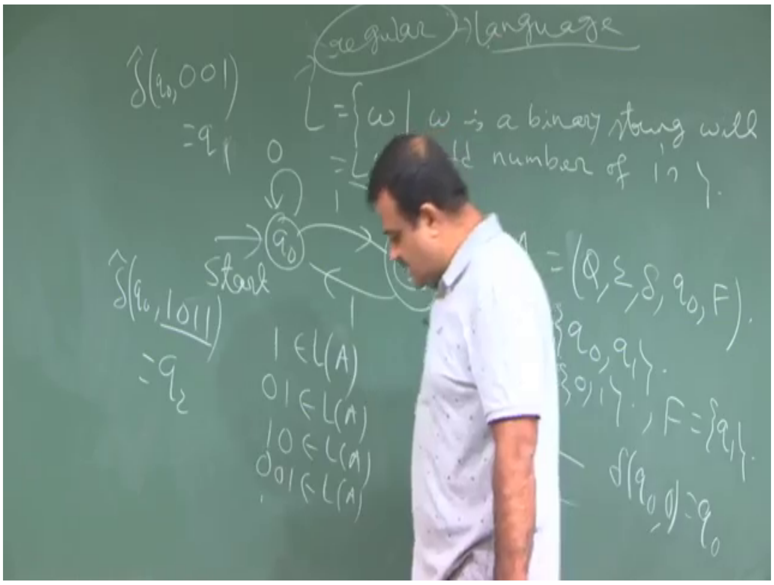
(Refer Slide Time: 15:41)



So, now the way we have defined this we can see 1 is accepted, 1 1 is not accepted, because if you put 1 1 so delta because 1 is accepted 1 1; so delta hat of q 0, 1 1 this is going where. So, here at q 0 you go to here, you go come back to q 0. So, this is not accepted 1 1 is not accepted. Now, 0 is the 0 is accepted, 0 is not accepted, because delta of 0 is remain 0 ok.

So, now two symbol like 1 is accepted, 0 is not accepted, 0 0 is not accepted, because it will be remain at this; 1 1 is not accepted. Then 0 1, 0 1 so delta hat of q 0, 0 1 so this is basically q 0, if we first we see a 0, it will be here, then we see a 1. So, it is accepted. So, 0 1 is accepted.

(Refer Slide Time: 16:45)



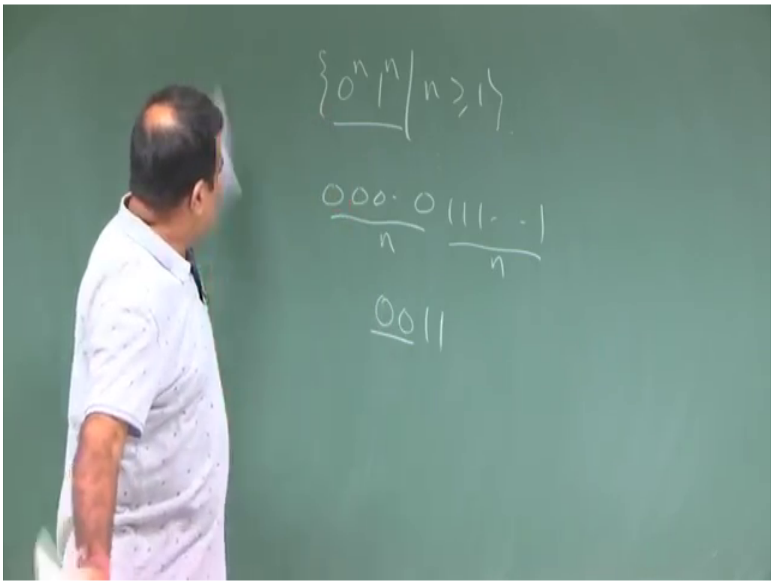
1 0 if we see a 1, will be is there and if you see a 0. This is q 1 ,because if you see a 1, we will go there and then we see a 0 will hop there. So, this is accepted 1 0. 1 1 is not accepted. So, now we check three, three means 0 0 1 this is accepted that is delta hat of q 0, 0 0 1. So, q 0 we see a 0 will be here, it is another 0 will be here, as I see a 1 will go there. So, this is accepted. So, 0 0 1 is accepted ok.

Like this we can check, this is the same language. If we have a even number of 1’s ,because if we just have even number of 1’s, we will just go there; if we have odd number of one, we go there we come back here, again we go there. To go there, we need to have another one. So, this is the way we construct this, is this clear. So, this is the same language as this. So, this is a L of A.

So, all the string which consists of even number of 1’s that will be accepted by this ok. Like, one like we can take 1 0 1 1. So, what is delta hat of q 0 this, we will start with q 0 we will go here; then 0 will remain there, then 1 will come back here, then again 1 will go there. So, this is even number one. So, this is q 2 accepted. So, this is accepted. So, this is the set of all possible string, set of all string which is having sorry which is having odd number of 1’s, odd means either one 1, three 1, five 1 like this which is having odd number of 1’s, these are all accepted. So, this is a regular language.

This language is regular, because we can construct a finite automata which is accepting the this is a regular language, which is accepting the all the string from this set, then it is called a regular language ok. Then we can have another example, like so now the question is, is every language is the regular? No, we will come later state that there are certain language which are not regular, I mean we cannot construct a we cannot construct a finite automata, which is a like this language.

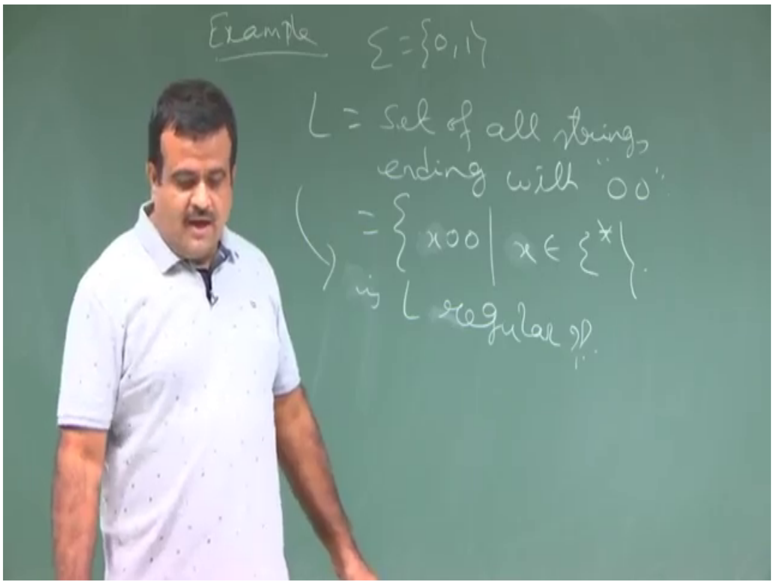
(Refer Slide Time: 19:55)



0 to the power n 1 to the power n; so, number of 0's is n and number of 1's is n. So, there are equal number of 0's and equal number of 1's, 0's followed by 1. So, for this language we cannot construct a finite automata DFA or we will see NFA is also not possible, because the DFA has no memory, we cannot count that how many 1's we have encountered, there is no extra there is no memory to count that.

So, how to count that we have encountered say if we have 0 0 1 1; so, we have two 0's then followed by two 1's. So, how to count how to store that number of 0; so, this is not possible in this structure. So, this is possible in pushdown automata so, will talk in more details on that later stage. So, it is not is it so this language is not regular, because there is no finite automata which can accept this. So, we will discuss more on this, in this type of language.

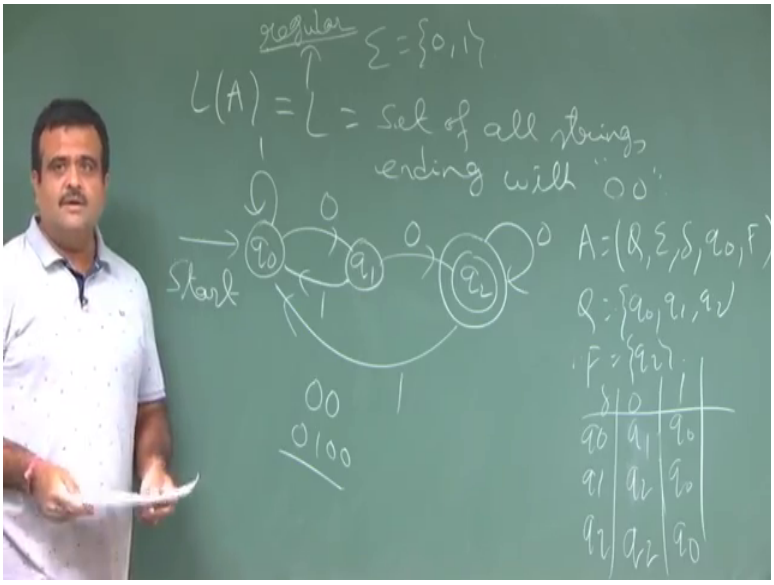
(Refer Slide Time: 21:11)



Let us take another example of regular language. So, suppose again sigma is binary stream ok. Now, suppose we defined a language like this set of all strings, all string binary strings ending with 0 0 ending with 0 0. Last two, last two bit as 0 0 ok.

So, this we can write it this way, this is w 0 0. w is belongs to star or x 0 0, x is coming from x 0 0, x is coming from sigma store, it would be epsilon also ok. Now, the question is, is this regular? Is L is regular? Is L regular? Ok, so this is the this is so to be regular what we need to have, we need to have a DFA finite deterministic finite automata which can accept this. So, can you try that yeah. So, can you try to have a DFA with this. So, let us just try to construct a DFA.

(Refer Slide Time: 23:03)



When it suppose we have q 0, q 1 two state is not enough, because we have to see 0 0. I mean we will see, I mean whether it is possible or not. So, this is the starting state. Now, if we see a 0, we can go to another state with hope that from again if we see a 0, we will go to the final state. So, this is the intuitive idea to construct a DFA like this.

Now, after reading 0 if you see a 1, so this is here. So, here if we see a 1 in half here, so and if we see a 1, we have to come back here with the hope that we will again sees the 0 0 like this. And from here that is ok, and from here once we reach there. Then if we see a 0 here, it can be here, because that is also, but if we see a 1, then you have to come back here. Then you have to come back here, because if we see a 1, then we need to have another two 0's for accept this. So, this is the constructs this is the finite state machine.

So, what is delta? So, what is our A; A is Q, sigma, delta, q 0, F. So, what is Q; Q is consists of three states, like q 0, q 1, q 2. And F is nothing but q 2 final state; and sigma is 0 1. What is delta, delta is like this. So, we have two input 0 and 1, and we have q 0 q 1 q 2. So, from q 0 if you see a 0, we will go to q 1. And if we see a 1, we are remain at q 0. From q 1 if we see a 0, we are getting sorry, this is yeah so this is q 1 right.

So, from q 1 if you are seeing a 0, you go to q 2 final state; but if we see a 1, we are again coming back to q 0. And then from q 2, if you see a 0, we are going to q 2 again; but if you see a 1 here, we have to come back to q 0, start again that thing ok. So, this is accepting the string like this, 0 0, because 0 0 is going there. Then the 0 1 is not accepting, then say a 0 1 0 0. So, all the string which is ending with 0 0, it is accepting.

So, L of A is nothing but this L. So, this is a this L is a regular language, because we do have a DFA which is accepting this language, which is accepting the all string of this language. So, if we can construct such a we for a given language, if we can have a finite automata which is accepting all the string of that language of that set, then we call it is a regular language. So, hence this is a regular language, because we can really construct the DFA which is accepting this.

Thank you very much.