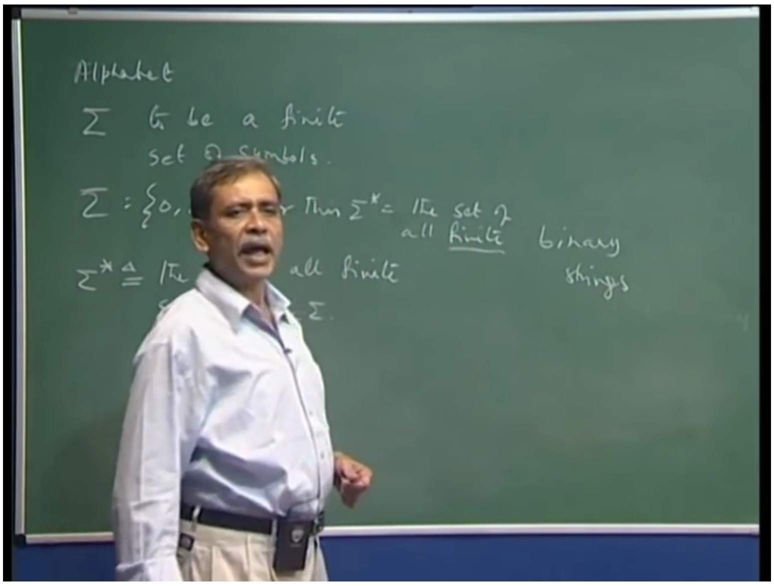
Theory of Computation

Prof. Somenath Biswas
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Lecture 2

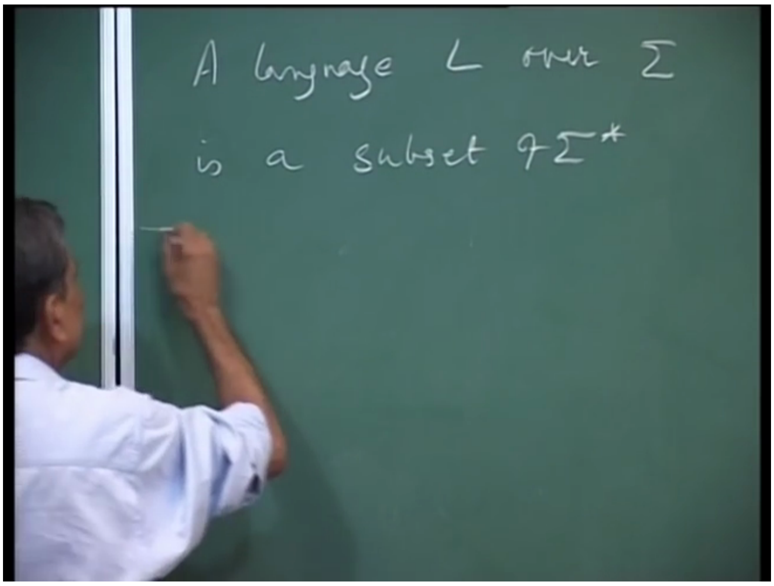
Introduction to Finite Automaton

(Refer Slide Time: 0:46)



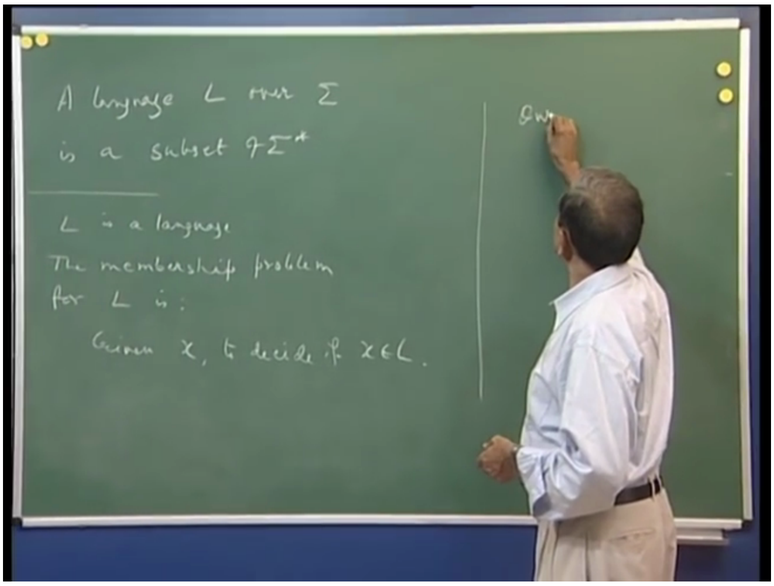
Recall we had defined an alphabet to be a set of symbols, a finite set of symbols and as an example we could have taken 0 or 1 or a, B, C, D through Z or any other set of symbols finite set of symbols that you choose to have then we also define Sigma star to be this was defined to be the set of all finite strings over Sigma. So in case of a sigma is just these 2 0 and 1 then Sigma star for this Sigma star is the set of all finite binary signs.

(Refer Slide Time: 2:30)



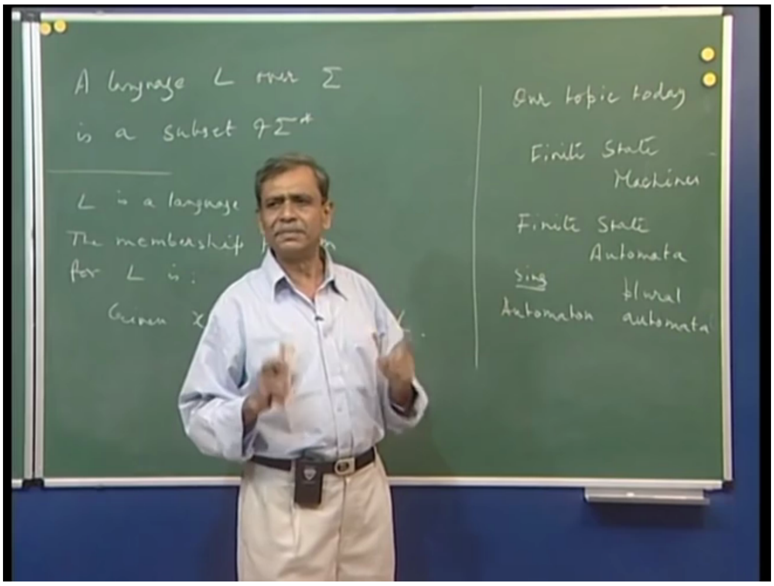
And another notion that we define last time was the notion of a language. We said a language L over the alphabet Sigma is a subset of Sigma star and throughout this course one theme that will continue is that we will consider various languages and see how we can solve the membership problem of these languages. Recall if I have language L then the membership problem for L is given a string x to decide if x is in L.

(Refer Slide Time: 4:31)



And today we will take a very simple very intuitive way of solving these problems for a class of languages. And we will do that in fact, all these membership problems that we will be solving we will have we will define some machines or grammars and these terms will be clear as we go by. So we will define something and we will show that this machine or this grammar we will be able to if you use that then you will be able to solve the membership problem of this language or that language.

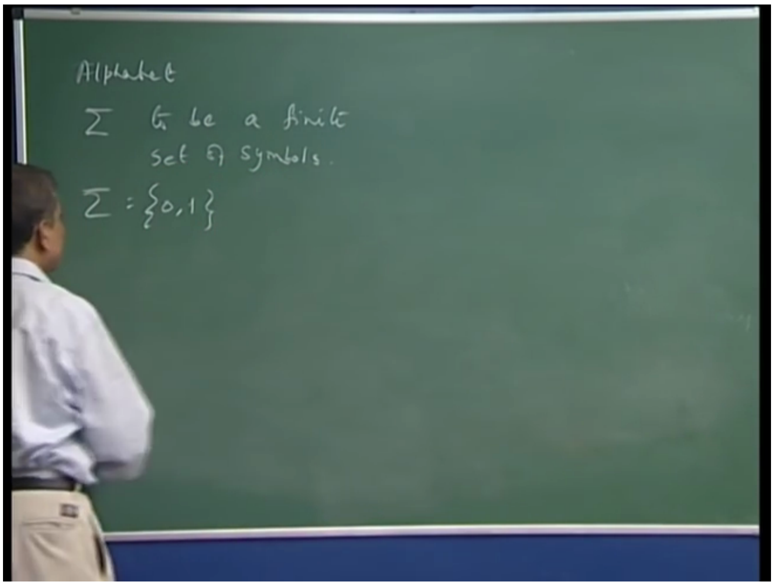
(Refer Slide Time: 4:49)



And as I said today our topic is finite state machines, machines when we talk of machines of course we do not really mean machines like lathe machine or drilling machine but machine in the sense things we do mechanistically and you will see that these are also kinds of very specified kinds of algorithms and best will be when we go through some example machines then we will be clear what we mean by machine possibly.

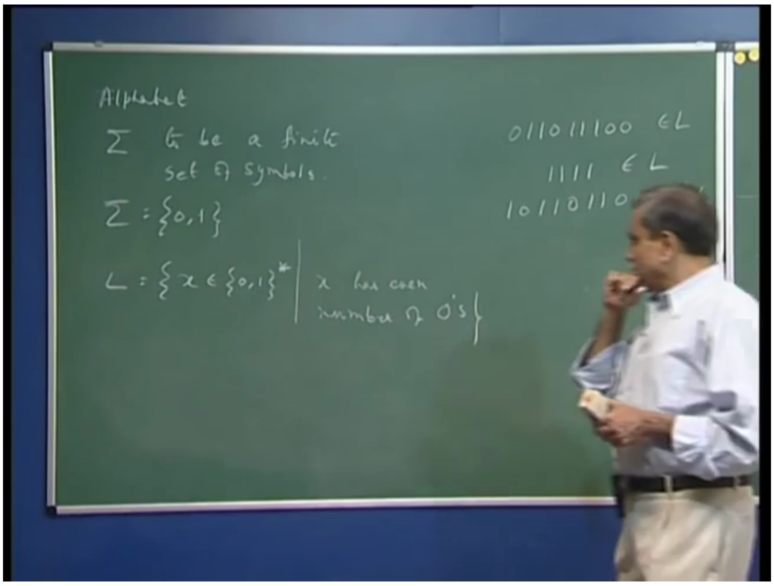
Incidentally these are also called which we are going to study today, finite state automata. The word automata is the plural of the word automaton, this is singular and plural form is Automato basically they mean what we mean machines in our context, alright. So let me introduce the notion of finite state automata or finite state machines through an example, I will introduce the notion of finite state machines or finite state automata through a very simple example.

(Refer Slide Time: 6:32)



So our example is going to be a language over this Sigma which is just 0 and 1 and let me say the language is the set of, let me write it in this way, all strings x which are in 0, 1 star immediately then I know I am talking of finite binary strings such that x has even number of 0's, okay. So this language is very clear that for example if I take this string, this is in this language reason it has one, 2, 3, 4 zeroes.

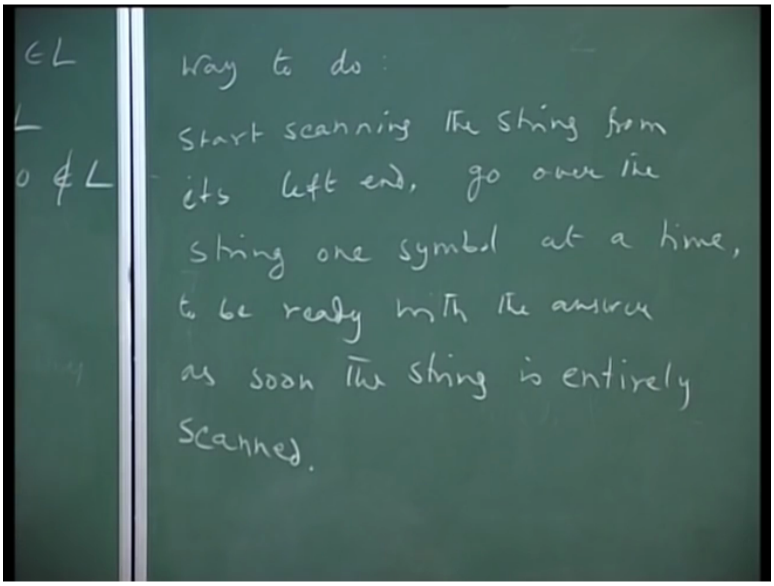
(Refer Slide Time: 8:31)



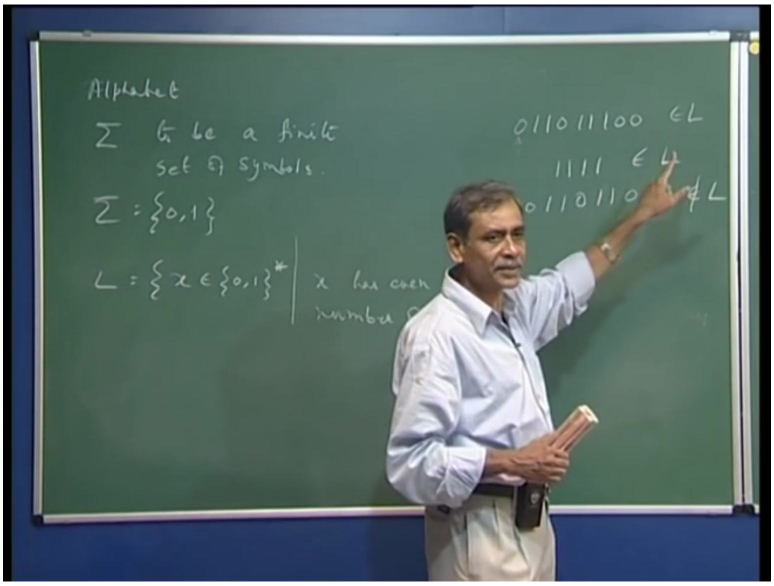
If I just take a string of 1’s this is also in the same language L because the number of zeros in this is 0 which is an even number. So this is an L, on the other hand if I take something like 101101100 this has 1, 2, 3, 4 and then suppose I put another 0 this is not in the language because this has 1, 2, 3, 4, 5 zeros which language this is a language we are discussing.

Of course you will see that this is a very trivial problem I will just count the number of zeros and if that that count happens to be even then I will say that string is in the language otherwise it is not. Yes, we could do it that way. However we will restrict ourselves, so that we motivate us towards this notion of finite state automata by saying that look the way you have to do this, make this decision. Solve the membership question of this problem; this language is this that when the string is given to you, you will start scanning from the left end, okay.

(Refer Slide Time: 9:18)

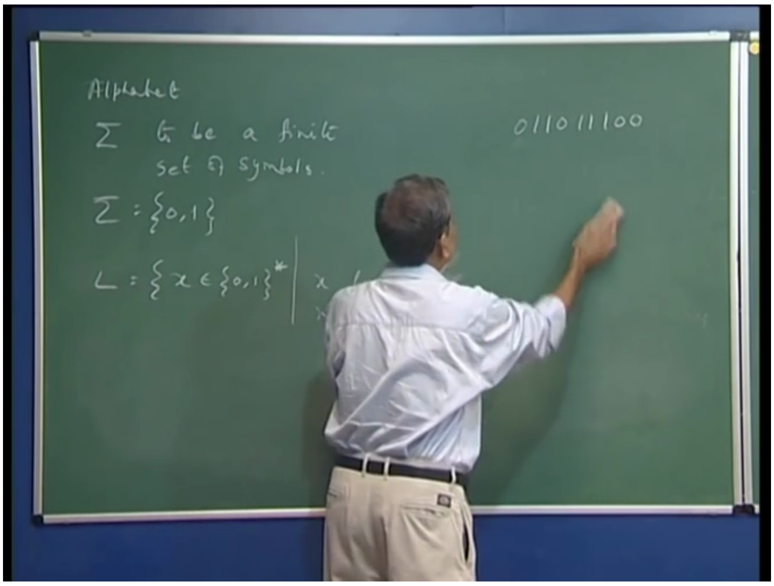


(Refer Slide Time: 10:18)



So let me write it down start scanning the string from its left end. Go over the string, one symbol at a time and when you come to the end of the string, when the string is over when your this scanning process as you are going from left end and you are going down down when you go the on the string that means when you have scanned the entire string your answer should be ready. To be ready with the answer as soon as the string is entirely scanned, alright.

(Refer Slide Time: 11:07)

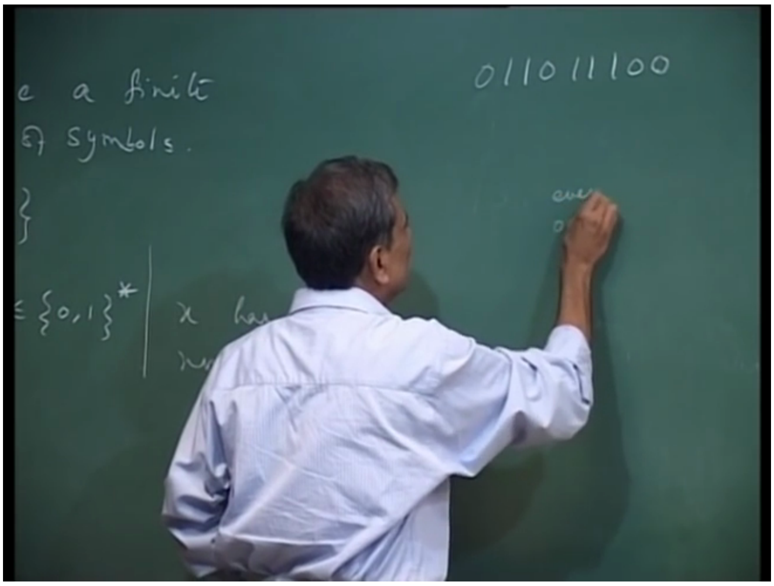


So think of this string, what we are saying is, we will start from the left end. So right in the beginning you will see 0 and then you will come to the symbol then this symbol than the symbol and so on by the time you finish scanning this string, entire string your answer should be ready and answer about what? Answering that whether this string belongs to the language L that means whether it has even number of zeros or not, thereby solving the membership problem for this language.

If we think a little bit, the way to do this would be, one way to do this would be satisfying this constraints that what at any given time, after scanning some prefix of the string I will remember whether the number of zeros have seen so far is even or odd, right? So in the beginning you have, when you started you had not seen anything, so you know imagine you are just about to start scanning the string from the left end.

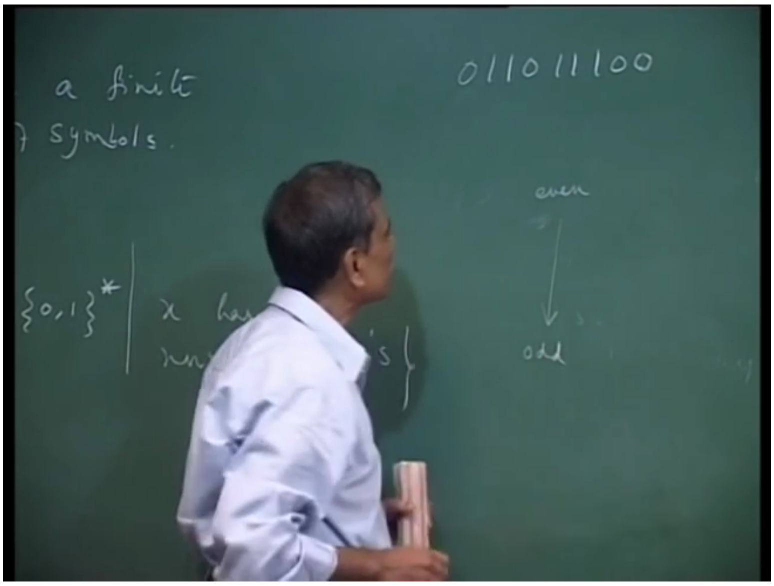
So therefore you have not seen any symbol at that time how many zeros have you seen? You have seen 0 number of zeros, right? So then that time you are state of the mind is that I have seen 0 zeros or even number of zeros then when you see the first symbol here, in this case it happens to be 0. So that when you finish scanning this symbol your state of the mind will be that I have seen so far an odd number of zeros.

(Refer Slide Time: 13:31)

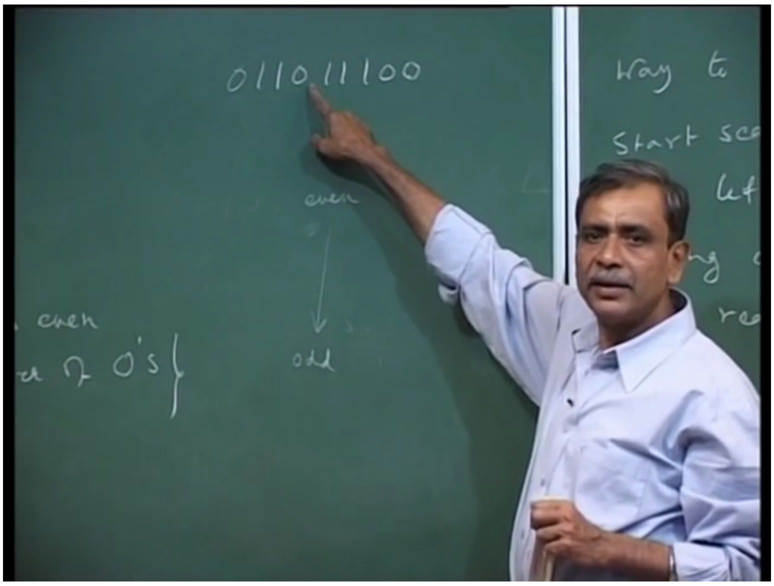


Then you will scan, so let me write this in the beginning you have seen even number of zeros that will be true for beginning of any string when I say beginning that means when I have not seen anything. Of course that means I have seen the empty string which has 0 symbols and therefore it has even number of zeros. So I will say initially the state of my mind as I am doing this, is that I have seen even number of zeros.

(Refer Slide Time: 14:08)



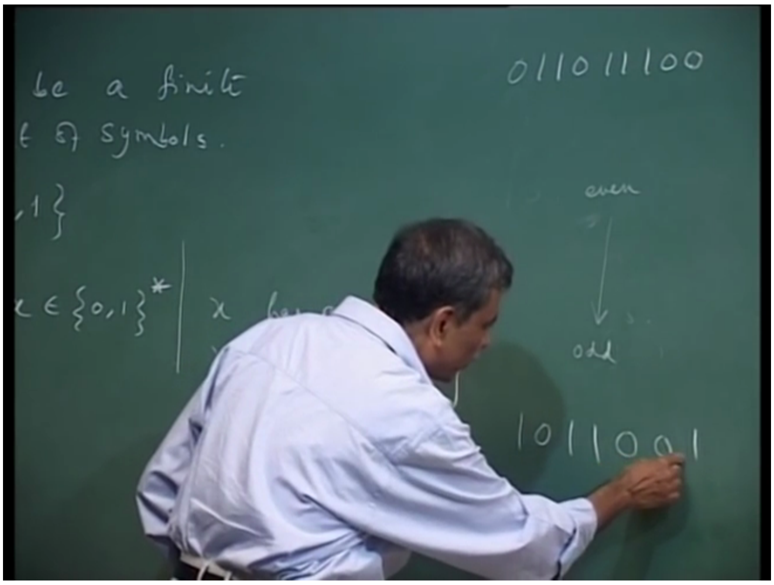
(Refer Slide Time: 14:35)



When I see this 0 then I say that look, now I have seen odd number of zeros that I see 1, my state of mind does not change because I have seen for this first part this 0 and 1, I have seen odd number of zeros that again another 1 comes even now I have seen only odd number of zeros and then when I scanned this one I say that, my state of mind if I am, if you allow me to say is that way.

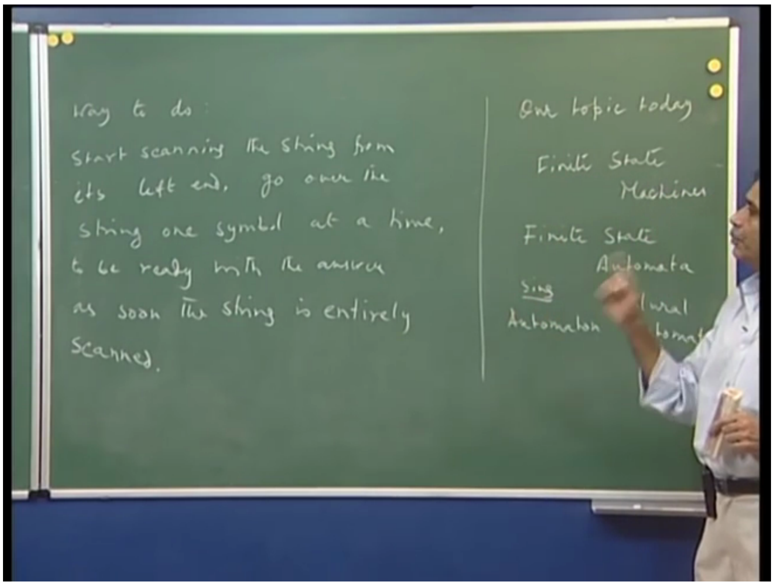
My state of mind is that now that I have seen even number of zeros. Again my state of mind does not change I have seen even number of zeros, I have seen even number of zeros. Now when I finish scanning this I have seen odd number of zeros, now when I come here I have seen even number of zeros, right? So I have as and when I finish scanning this entire string I know that number of zeros in this string was even.

(Refer Slide Time: 15:50)



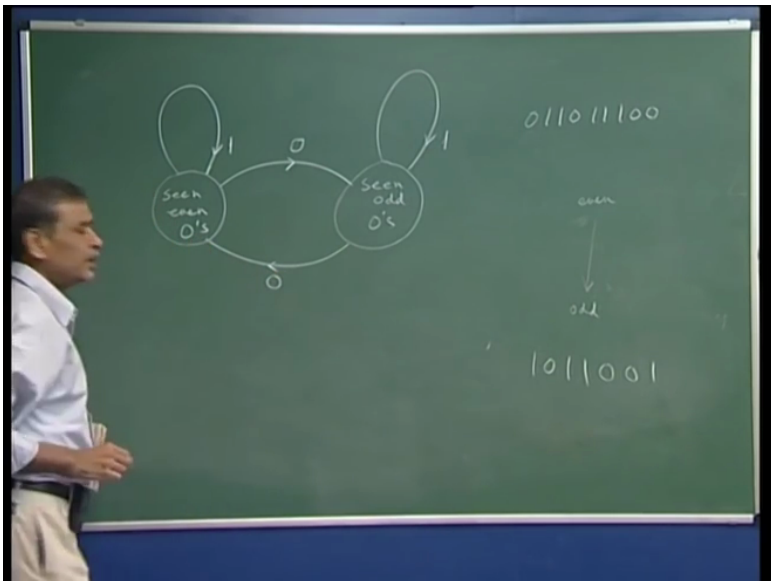
On the other hand supposing my string was 101 1 0 0 1. So again if we do it that way in the beginning I have seen nothing, so even number of zeros I have seen, even number of zeros I have seen. Here when I finish scanning the symbol odd number of zeros, odd number of zeros, odd number of zeros, after this even number of zeros, odd number of zeros, odd number of zeros.

(Refer Slide Time: 16:00)



So it is possible to satisfy these constraints and be able to decide whether or not the string which was given to me has even number of zeros or odd number of zeros.

(Refer Slide Time: 16:31)



So let me say what we were doing through a picture or a diagram. Let me say that my mind has 2 states. So far as this problem is concerned in this state I have seen even zeros, here have seen odd zeros, right? And how do I go from one state to another because you see a symbol came and when I after scanning that symbol I may depending on whether it was 0 or 1 I may change the state of my mind.

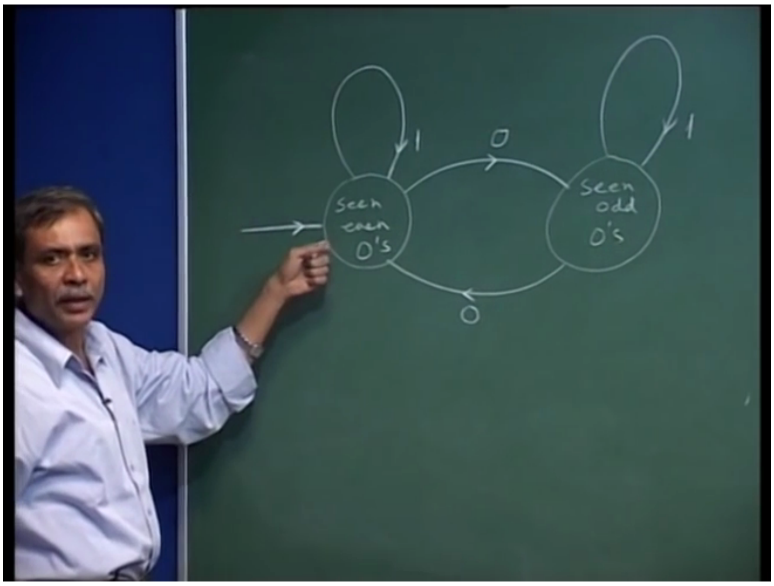
So for example in case you are one comes, I am here that is I have seen even number of zeros and 1 comes, should I change my state? No because even then I have seen only an even number of zeros. So what this means is, suppose I have scanned so far prefix of the string which has even number of zeros and then I scan under the symbol which is 1, still clearly I have seen only even number of zeros.

So therefore I come back to the same state and you will see this if I use in this manner. If 0 comes then if I had seen so far even number of zeros I will come here my state of mind will see will record that I have seen odd number of zeros and here if I see a 1, I will remain in the same state of mind and if I see a 0 here then of course previously I had seen so far odd number of zeros, one would 0 came therefore I have seen even number of zeros.

So I move back and forth from state to state, there are just only 2 states in my mind to do this problem suffices that we keep just these 2 states of mind and as and when a symbol comes I may or may not change my state depending on what the symbol was, right? Now you seen the beginning I said that when this process of scanning the input string starts where am I? Which of these 2 states?

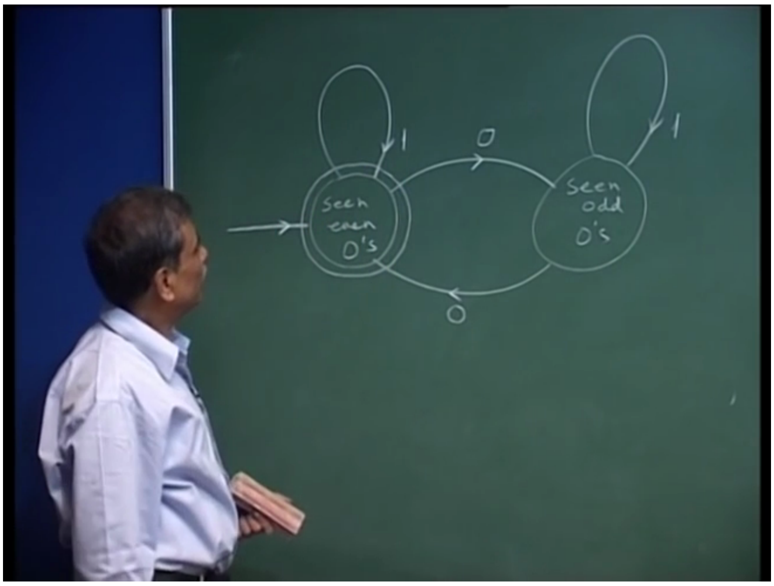
I will be here because I had not seen anything, so therefore I had seen 0 zeros, so I am in the state and so this state is where this whole process begins.

(Refer Slide Time: 19:42)



So let me identify this state this manner and arrow coming from nowhere, right? In this picture it means in the beginning initially we are here, in this state and when I end I will say the string is in the language, remember the language was the set of all strings which has even number of zeros, if after finishing scanning the string, if I happen to be in this particular state. So the state in which or the set of states in which if I am in after scanning the entire string.

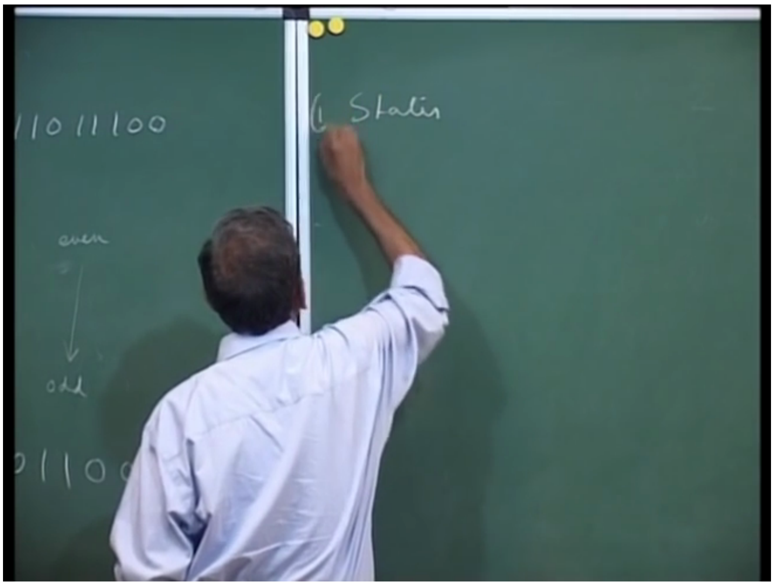
(Refer Slide Time: 20:46)



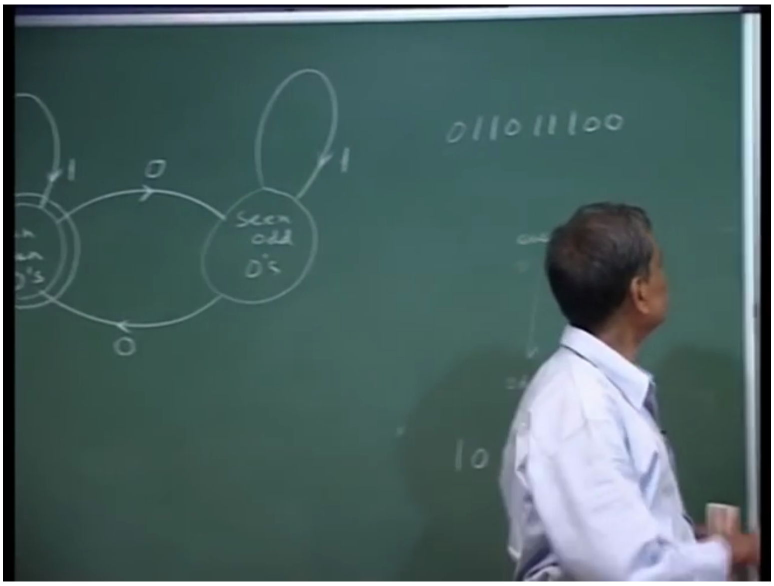
If those states are such that, they signal to me that the string is in the language, such states also need to be marked specially like as in this case, I will just put distinguish such states by making double circle, okay. You will agree that whatever it is, this picture tells me the way of deciding whether a binary string has even number of zeros or not in this manner when is starting my scanning from the left end going symbol by symbol and by the time I end scanning the entire string I have my decision.

So let us just see the components of the kinds of things that we have seen in this very simple example we had these 2 things and I have been telling that these are my states of mind for this problem.

(Refer Slide Time: 21:56)



(Refer Slide Time: 22:23)

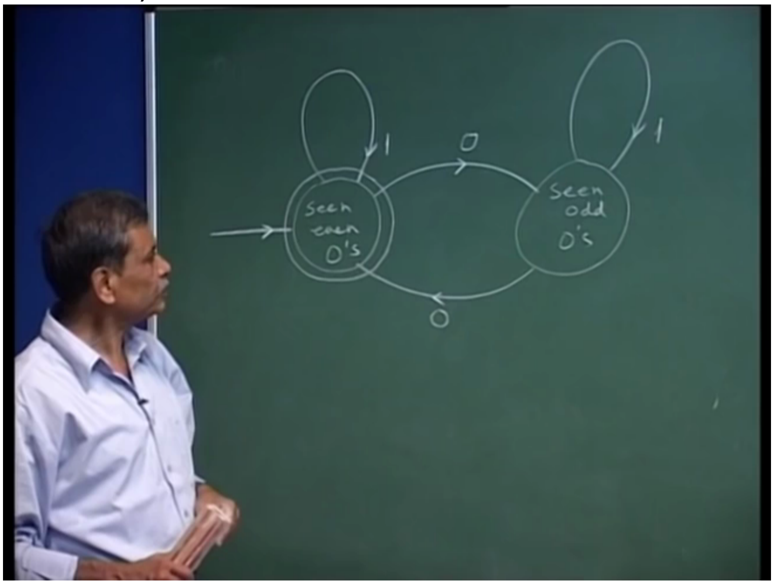


So we have some states, component 1 of this picture is a set of states and how many states? Well it does not matter, in this case there are 2 but such a set I will if I, I will make use the notation capital Q that will denote the set of states in pictures like this then what did we have? In this picture if you see I had these 0's and 1's and this marked here 0, 0, 1, 1 and they are what?

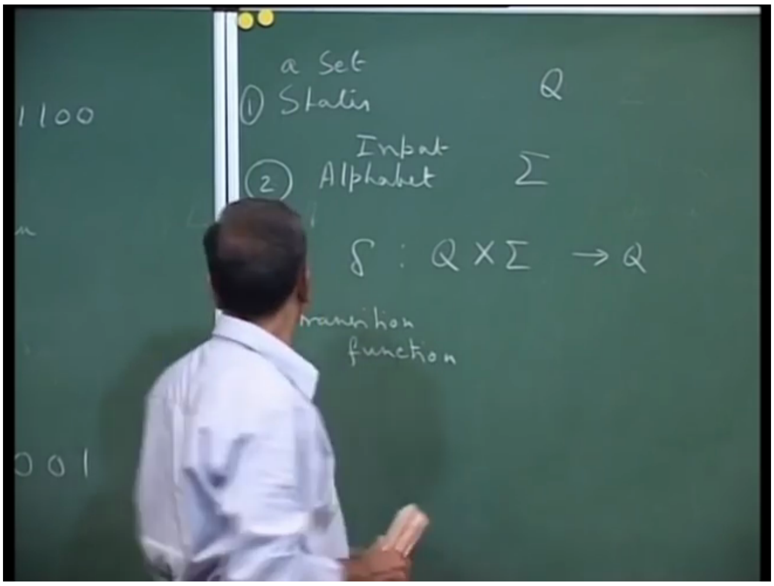
They are the symbols of my alphabet, so of course we will see many more examples of this kinds of diagrams but even for this diagram it is clear that what I put next to an arrow is a symbol of my alphabet when you see if you are in some state then clearly any of those symbols from your alphabet can be the next symbol, right?

We are in this state, 0 may come or 1 may come, right? So here we will have this alphabet which is the input, right? Input alphabet then as usual we will call it capital Sigma then a very important thing this is a rule for going from state to state, what do we do? I mean what is this idea that we are in some state, a symbol comes like we are in this state the symbol 0 came then we moved to this state, so this is called state transition.

(Refer Slide Time: 24:09)

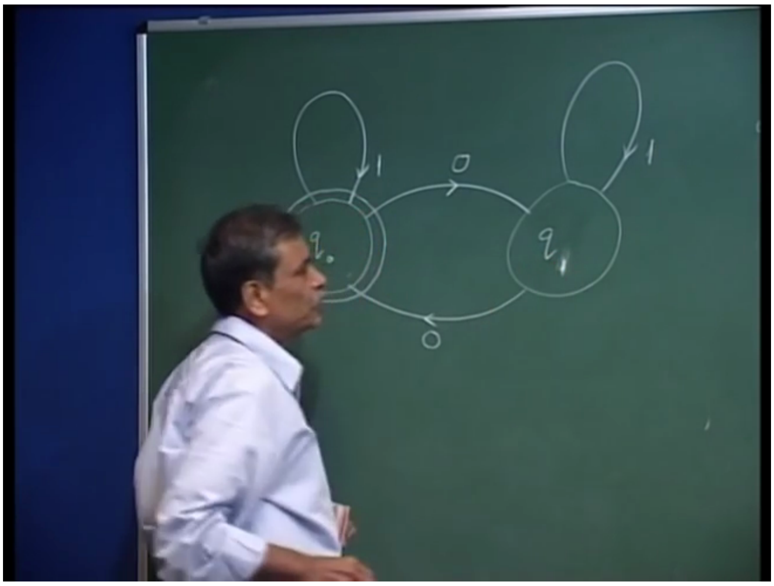


(Refer Slide Time: 24:56)

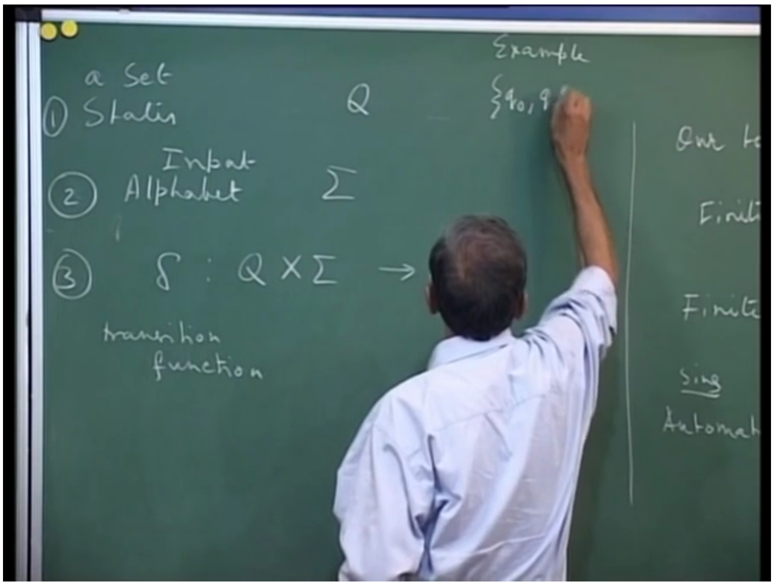


Now whatever is in this picture, I can think of that I have a mapping Delta, this is called transition function and this transition function what is it doing? What is it telling me? That if I am in some state and a symbol comes which is the next symbol of the input then which is the state I will go to? So this is a mapping whose domain is the product of the set of states and the set of symbols and the range is the set of states.

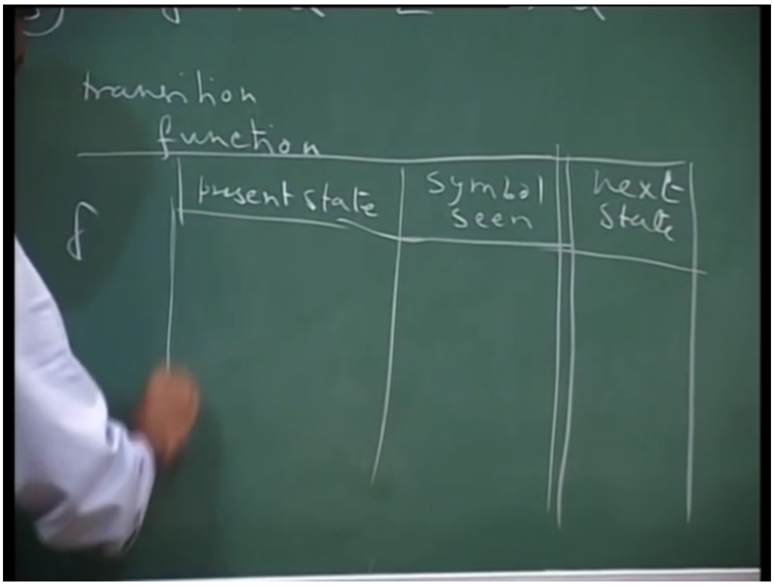
(Refer Slide Time: 25:20)



(Refer Slide Time: 25:32)

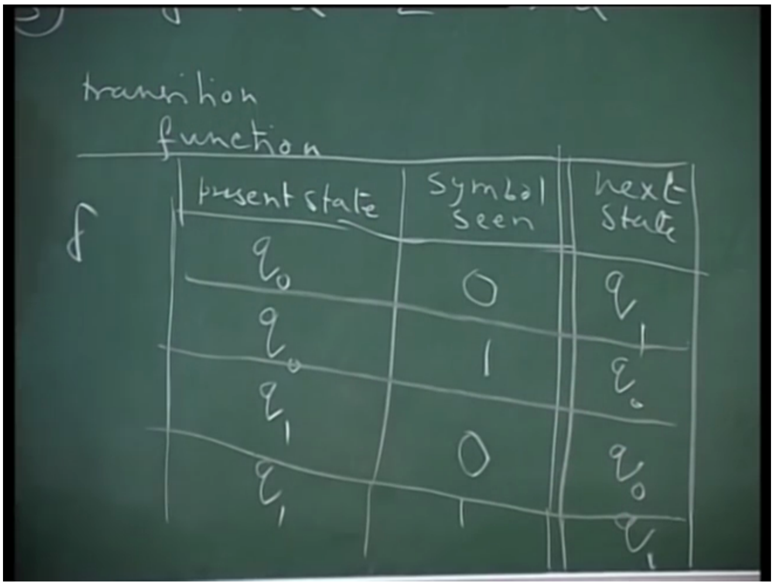


(Refer Slide Time: 26:18)



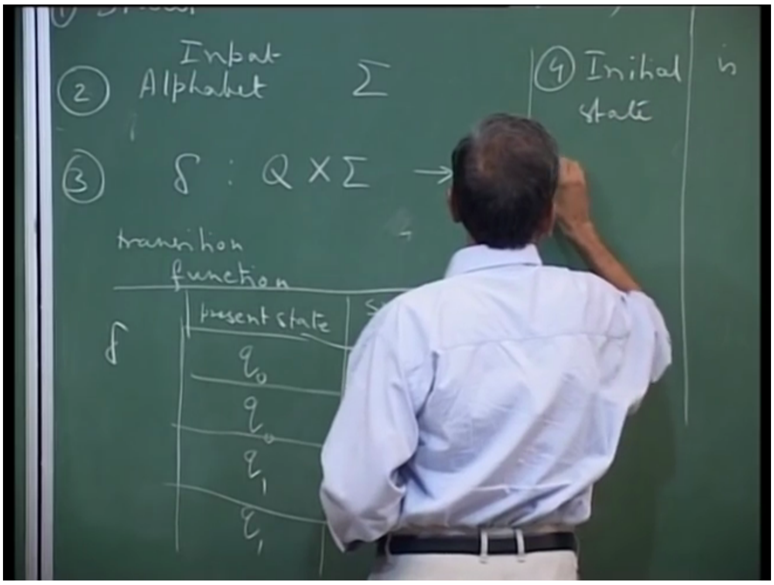
Now let me name these states this way. Now let me say this is, this state what I have written is for my own understanding but in general I will give some names like q0, q1. My 2 states in this example Q as capital Q as q0 and q1 and your Delta will look like this, something like you can imagine that Delta is a mapping which is taking the present state and the symbol seen and here I have next state, right?

(Refer Slide Time: 26:30)



So if the present state is q0 and if I see 0 then next state is q1, if the present status q0. If I see 1 I remain in the same state q0, if the present state is q1 I see 0, I go to the state q0 and if I am in the state q1 and the symbol is 1 and then we go to the state, we remain in the same state, right? This completely specifies the transition function as we have used in this example.

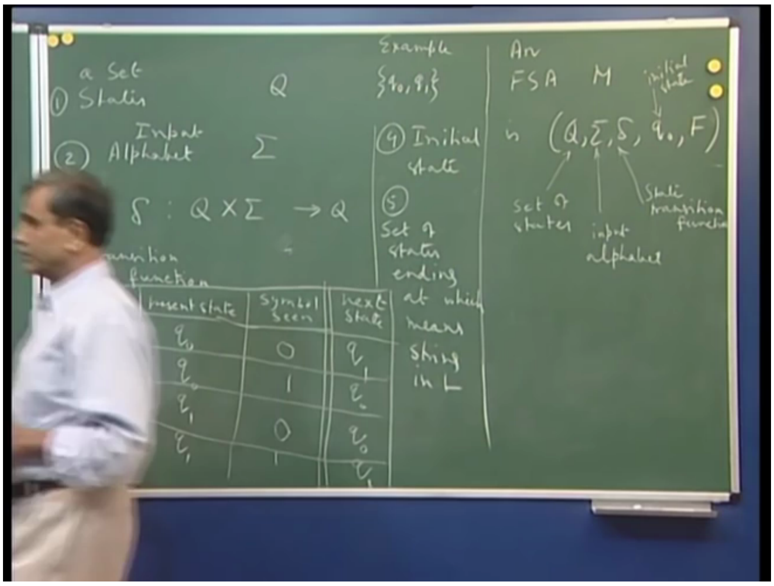
(Refer Slide Time: 27:41)



Now let us just analyze this, we will say a finite state automaton FSA and FSAn is in this manner, what are the components it has? Q, Sigma, Delta but you there are some other things in this diagram which so far I have not put down. What others are 2 things? One is where I started from? So here in this diagram this arrow coming from nowhere represented that state I was initially.

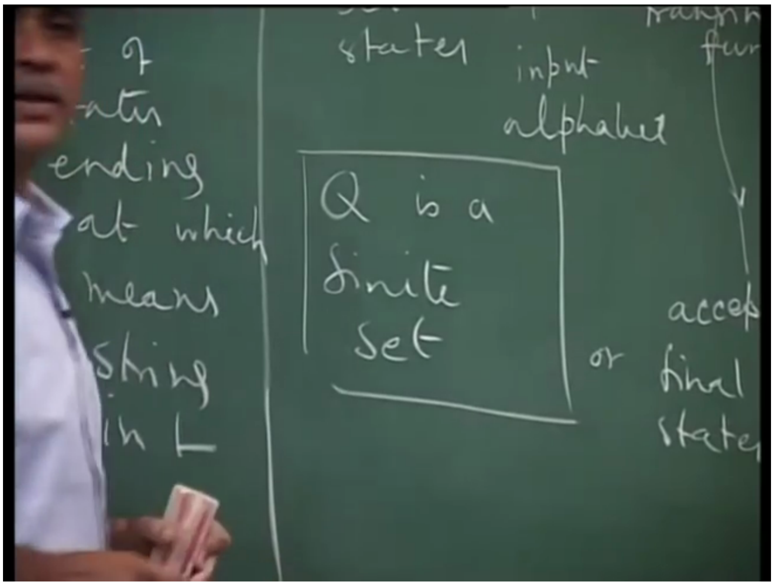
So my fourth component is initial state and the final component is of that in this picture like in this case this state q0 was specially marked to signal that if I end here that means the string is in the language. So there can be a number of states which are like that, all those states, so let me write this here set of states ending at which means string in language, okay. So there are 5, really if you look at this that there are all these 5 components were there and nothing more.

(Refer Slide Time: 30:33)



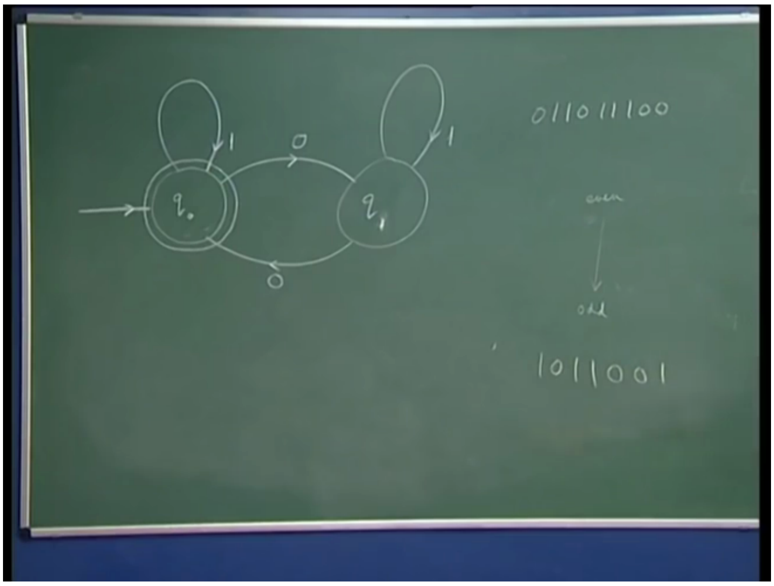
The set of states, the input alphabet Sigma are transition function which is specified that diagrammatically here Delta and then an initial state we call it q0 and this set of states ending at which means string is in L, there is a name for this kind of set states they are the set of final states or accepting states. So let me write these components, set of states, alphabets, input alphabets, this is the state transition function, this is the initial state.

(Refer Slide Time: 31:40)

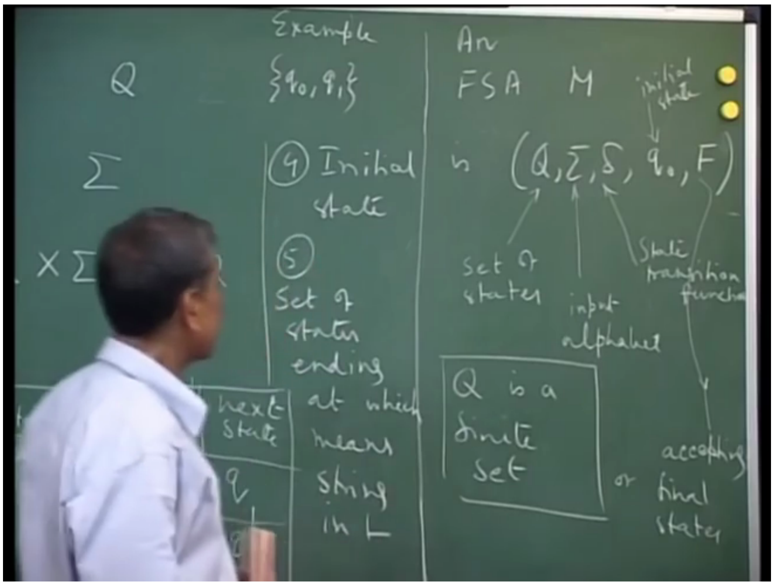


One initial state is not a set. It is just one unique state where you start and this is the set of final state or accepting state. See these set of states ending at which means string is in L, this F let me call it as accepting or final states. One important thing to say an FSA there is this one finite is there, what does this finite mean? This means very importantly Q is a finite set.

(Refer Slide Time: 31:48)

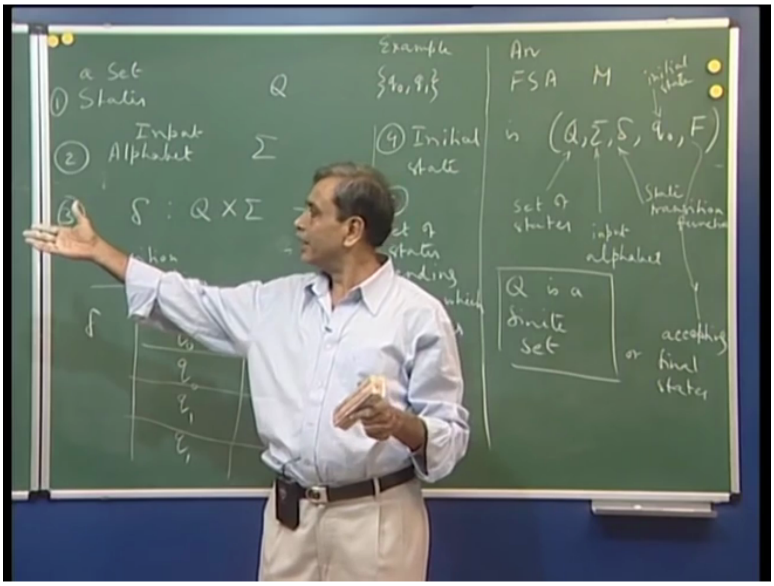


(Refer Slide Time: 32:24)

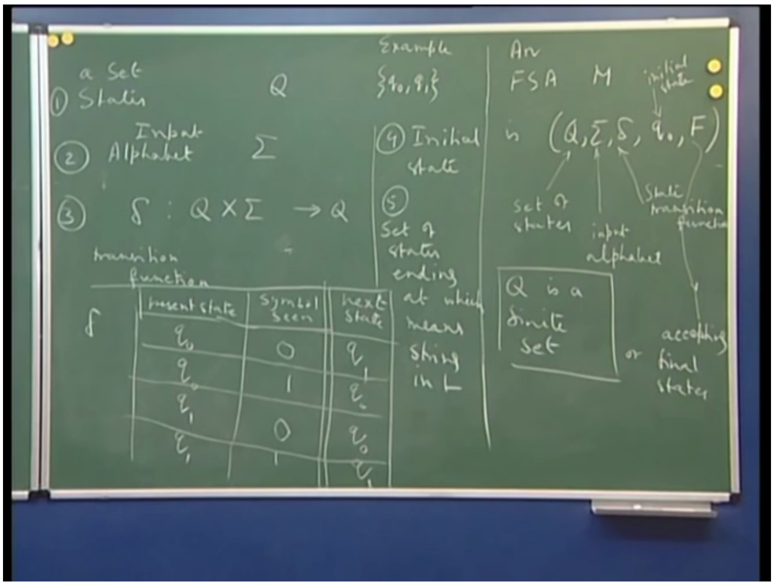


In other words this machine may have maybe 1,000,000, 1,000,000,000 states you know not this particular one but in general but the number of that number of states, total number of states is something finite. So in other words this I am trying to say is an FSA or finite state automaton M is something which has these 5 components, set of states, an input alphabet, state transition function, the initial state and the set of accepting states.

(Refer Slide Time: 32:45)



(Refer Slide Time: 33:18)

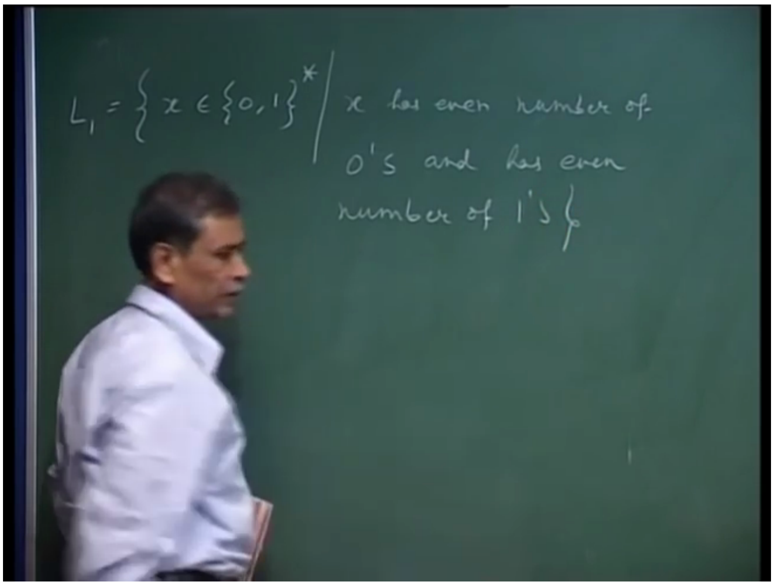


And you can see some of the relationships that we have, we have already mentioned them one is that of course we say that Q, cardinality of Q is finite, so Q is a finite set, Sigma is of course is also finite, is a finite set of symbols. Now what was Delta? Delta was a map from Q cross Sigma to Q and see that Q is finite, Sigma is finite. So this map can be specified by a table like this.

If this had 100 states, Q had 100 states there were 10 symbols then how many entries would I have the stable? This can take 100 values, this can take 10 values and so total number of entries in this will be 1000, right? For every state symbol pair I should tell what is the next step is? So Delta is a mapping and therefore you can see this is also something very finite and depending on the size of Q and Sigma.

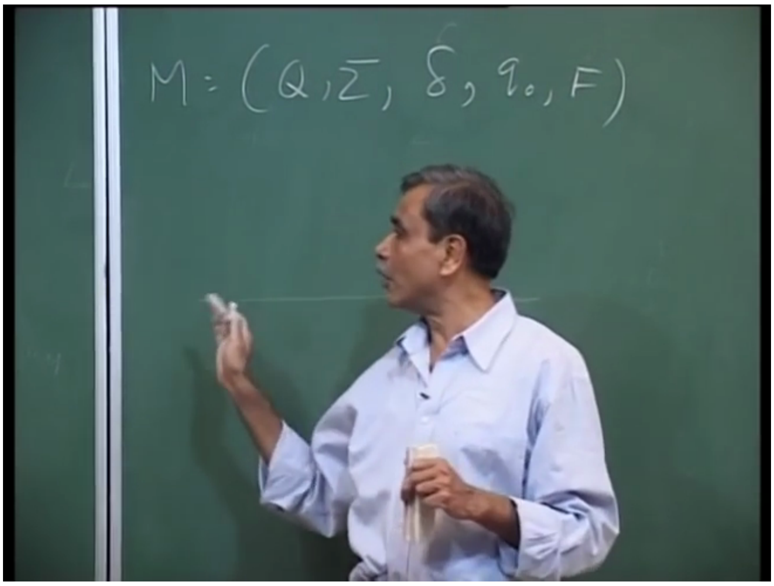
The size of Delta is there I mean that is thereby determine and q0 the initial state is an element of Q and F the accepting or the final set of final states is a subset of Q, okay, right? So now all these may at this point of time seem very artificial because there are very trivial problem that we have did and from there we are generalizing but let us see a few more examples to fix this idea of finite state machines.

(Refer Slide Time: 35:01)



So my next language that I will consider for which I will provide yet another finite state automaton is slide General generalisation the same problem. So my next language is L1 which is also a language over the binary alphabet. So this is set of all strings over the binary alphabet. So immediately I am saying it once more that immediately I know an x is a finite binary string and now x has even number of 0’s and has even number of 1’s.

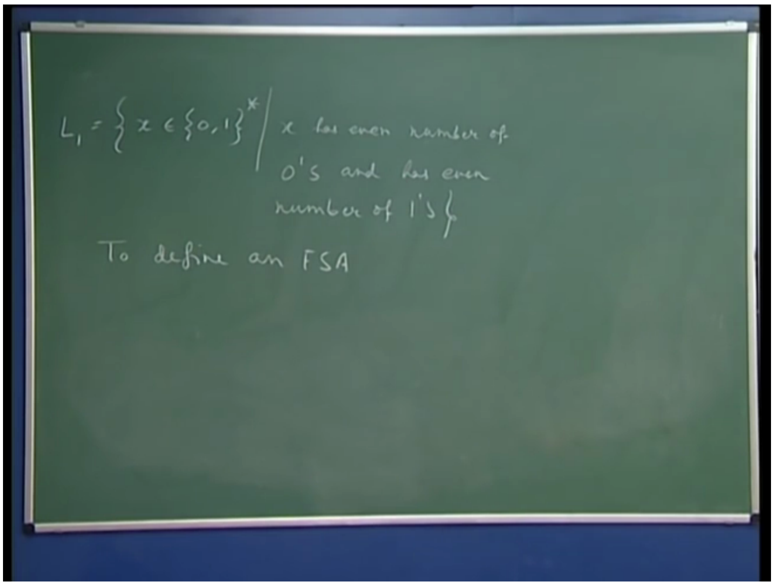
(Refer Slide Time: 36:38)



Let me rub this out but we should remember what we are trying to do? We are trying to give a finite state automaton which will solve the decision problem of this set L1 and I specify a finite state automaton by providing some machine M which has these 5 components Q, Sigma, Delta, q0 and F and the problem at hand is, can I give a machine M which will accept or I am using a word which will I will define a little later.

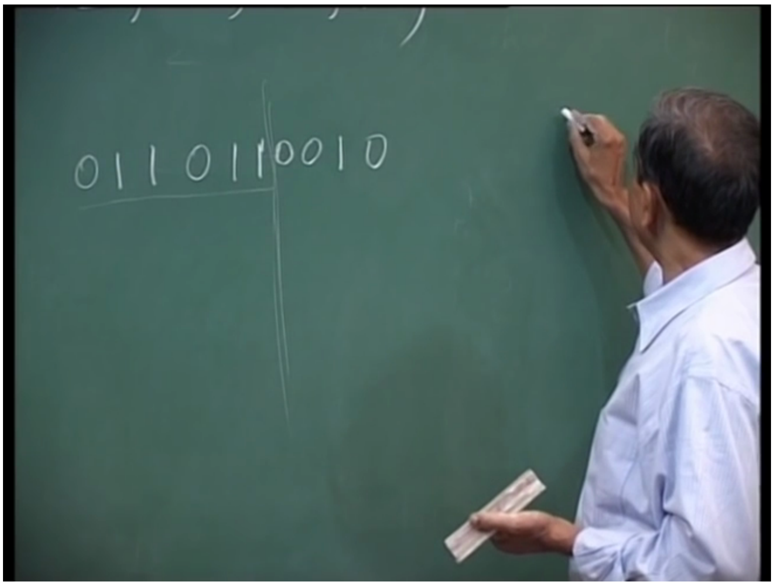
But basically what I am trying to say? The same thing that I want to give a finite state automaton such that this when this automaton finishes scanning this. Remember the game is that whatever string is given, we start from the left end supposing this is the string start from the left end, scan symbol by symbol and then when I come here my decision is already made, decision in this case whether or not a string that I have seen so far seen, when I have seen it entirely whether it has even number of 0's as well as even number of 1's.

(Refer Slide Time: 38:15)



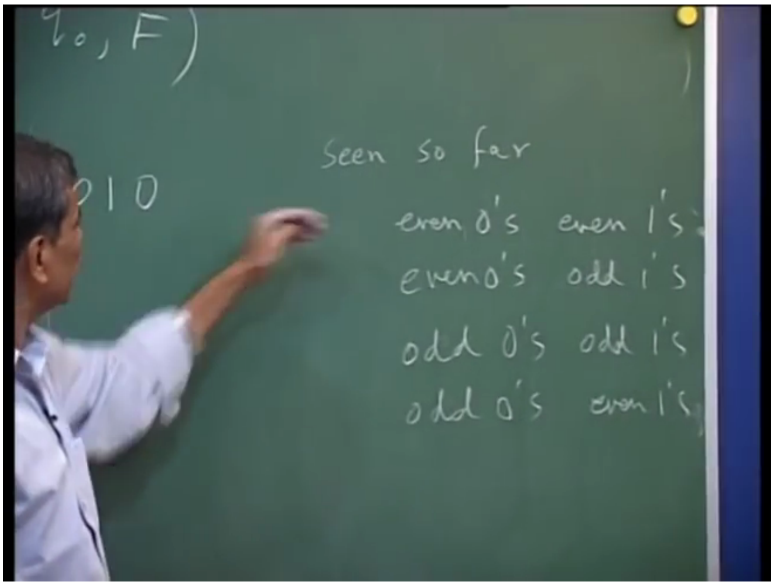
And we would like to do it in the same style, so let us think if we again go back to our old picture that for solving this problem. So our problem is we have this language L1 and we would like to define, to define an FSA finite state automaton which will be such that when that automaton finishes scanning the string which is given to any string which is given to it as an input, this automaton will be in a state which will signal to me whether this string has even number of 0’s and even number of 1’s, right?

(Refer Slide Time: 39:13)



So we kind of extend our old idea, let us think that what are the so-called states that I need to be able to do this. So again the same idea will generalize, we will see that look supposing I have a string like 0 1 1011 0 0 1 0. So if I have come here that is I have seen a prefix of the string I will remember whether I have seen even number of 0's and even number of 1's or even number of 0's and odd number of 1's.

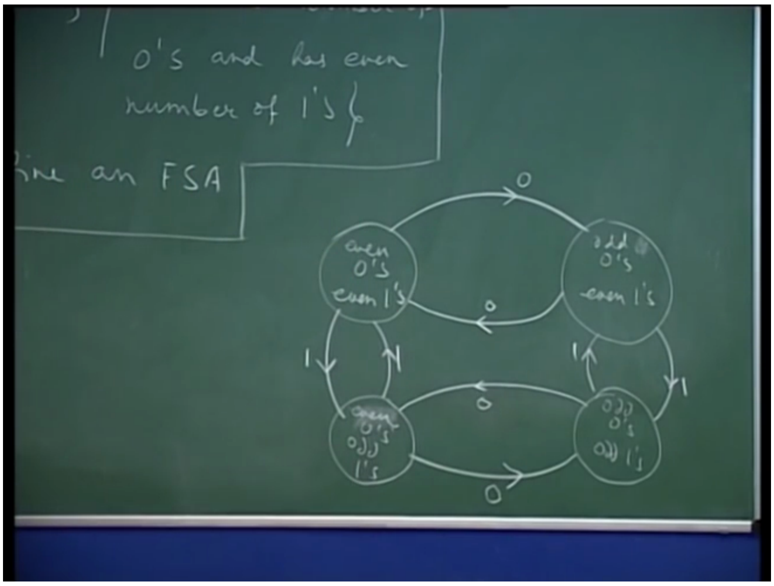
(Refer Slide Time: 40:37)



So let us see what are the possibilities? Supposing I am here what are the possibilities? That could be either I have seen so far even 0's even 1's, even 0's odd 1's, odd 0's odd 1's and odd 0's even 1's. You will agree that any binary string will fall in one of these and what more? You see for example here this string has even number of 0's and even number of 1's and now I see the next 0.

Then what happens? It means immediately I have seen and odd number of 0's when I finish scanning this particular symbol I have seen and odd number of 0's and since number of 1's did not change I have seen odd number of 0's and even number of 1's.

(Refer Slide Time: 41:25)



So we can actually do this way that let me have these 4 states even 0’s even 1’s and odd 0’s even 1’s, odd 0’s odd 1’s and here even let us say odd odd 0’s odd 1’s. So what is the one which is left? Even zeros even ones, odd zeros even ones, odd zeros odd ones, even zeros and odd ones, right? So here 0 comes, we are in the state of mind 0 comes then what happens?

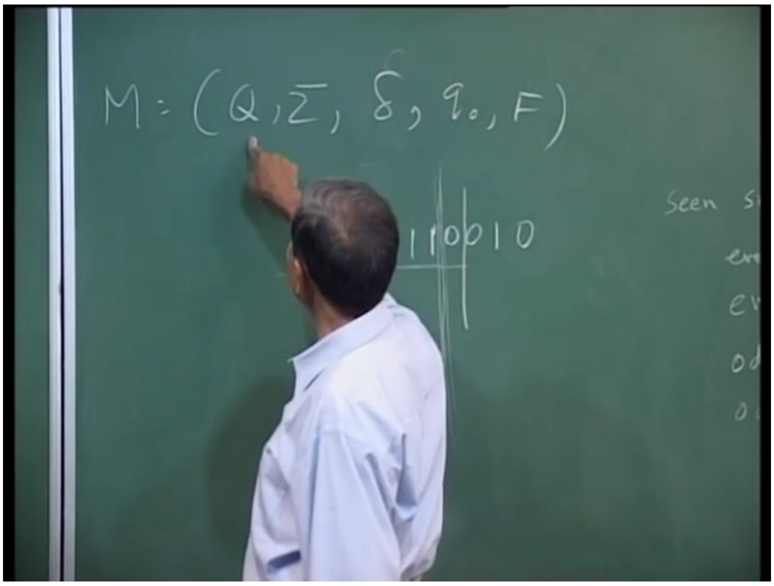
Clearly this is a situation, similar is the situation I had scanned the string up to this have seen even number of 0’s and 0 came. So clearly now if 0 comes, the number of 1’s do not change so still I will have seen only even number of 1’s but now because of the 0, total number of 0’s I have seen is odd and you can see from here, if we have seen odd number of 0’s and even number of 1’s and another 0 came here then I would have my state of mind would be, that I would have seen even number of 0’s even number of 1’s, right?

And what about here? If I see a 0, of course we have taken care of and here if I see 1, what happens? Even number of 0’s, the number of 0’s will remain 0 but if I see a 1 that means number of 1’s I would have seen will be odd. So even 0’s and odd 1’s is the state I will go to if I see 1 here. From here if I see 1 what happens? I will go here clearly, right? Now in this state, this means so far I have seen odd number of 0’s and odd number of 1’s.

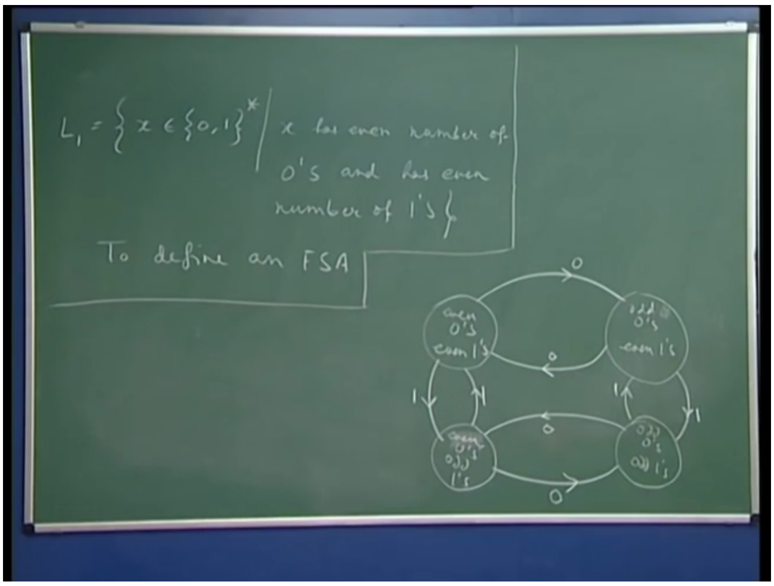
If I see 0 then I will come here, see I have seen so far odd number of 0’s and odd number of 1’s. One more 0 came, so therefore I would have seen now, even number of 0’s and odd number of 1’s and if I see 1 in state of mind or whatever you call it then I will go to this state, see I had odd number of 0’s, odd number of 1’s, one more 1 came, so odd number of 0’s even number of 1's and now you can complete the picture even number of 0's odd number of 1's if 0 came then I will go to odd number of 0's and even odd number of 1's.

And here if, so this is 0 and this is this arrow will be there because if there are even number of 0's and odd number of 1's and then 1 came and then I would be in seen even number of 0's and even number of 1's.

(Refer Slide Time: 45:47)

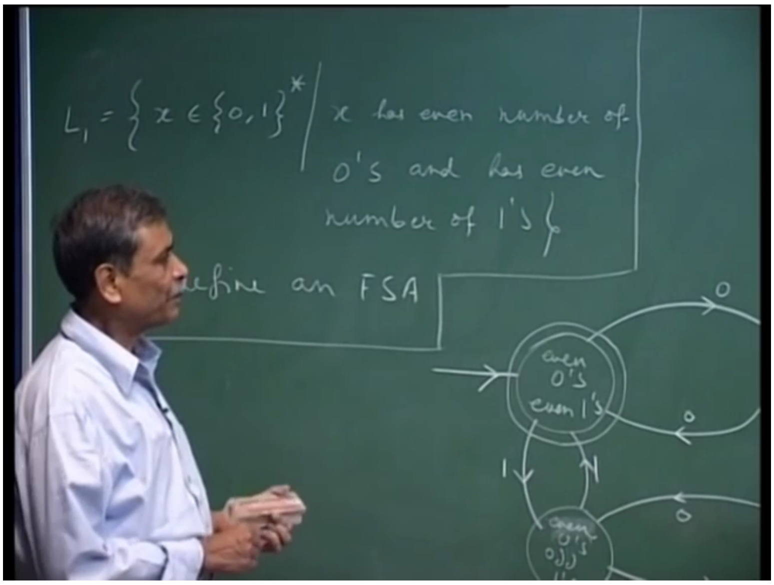


(Refer Slide Time: 45:53)



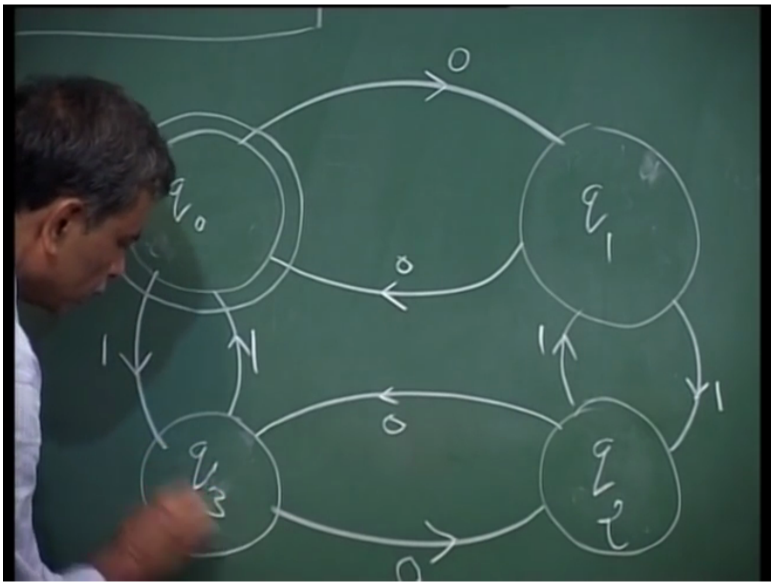
So of these I have specified, so this is 4 states Sigma is of course 0 and 1 these diagrams clearly tells me what Delta is, the transition function but what about the initial state? Clearly when I have initially I have seen the empty string which has even number of 0’s and even number of 1’s, so this is the initial state.

(Refer Slide Time: 46:34)



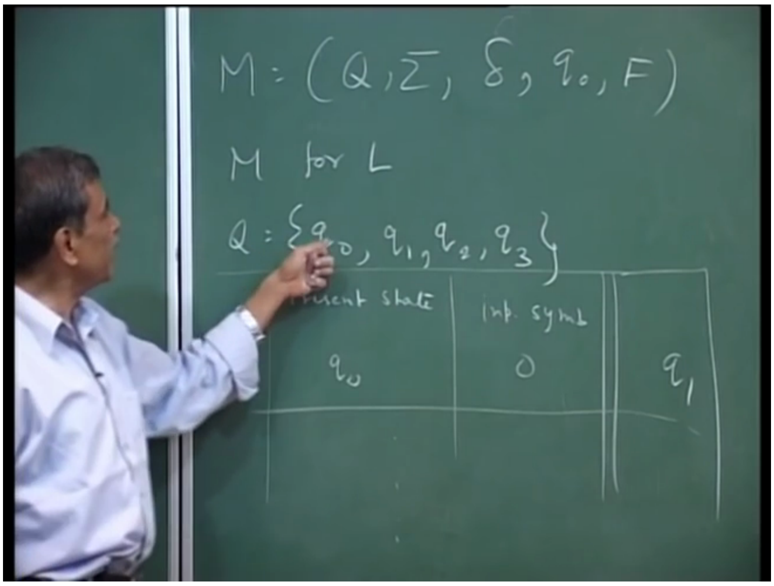
Which are the states in which if I am in at the end of the scan and the string is in the language it has even number of 0’s and even number of 1’s you can see this is the only state which should be marked like that.

(Refer Slide Time: 47:20)



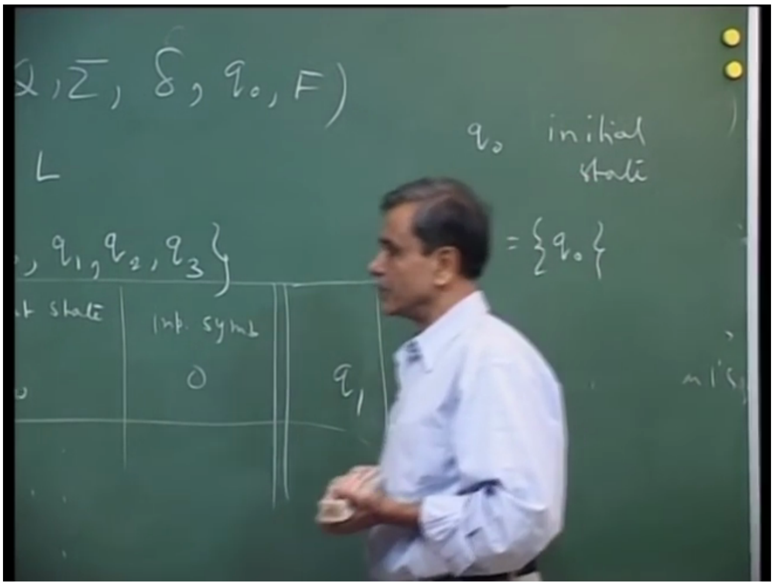
So for this example I have, now again I will read these things out because this was for my convenience I will say this is q0, this is the state q1, this is let me say q2 small q2 and this is q3.

(Refer Slide Time: 47:27)



So the machine for L1, M for L1, one machine it maybe many others but at least this one has Q is q0, q1, q2, q3 Delta is you know we can just translate this picture into a table like the one I did before for example here again I will have now let me let me give you couple of lines of this, if the present state is q0, present input symbol is 0 then my next state is going to be q1.

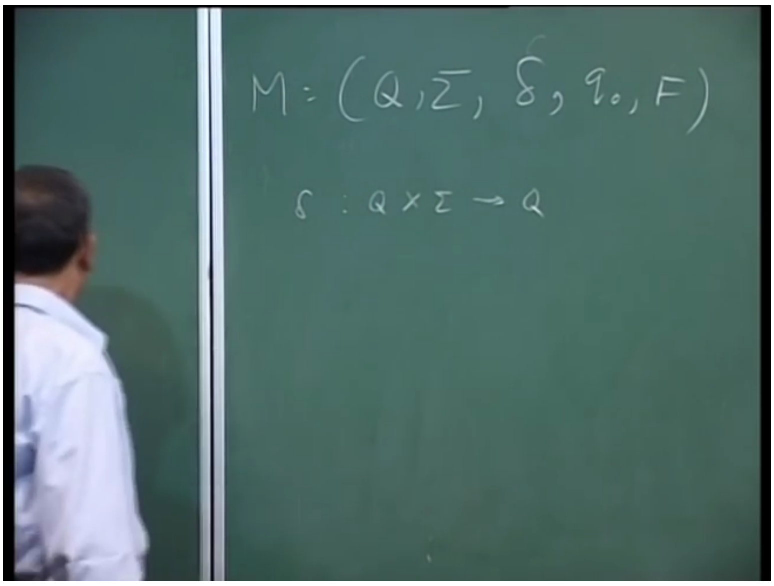
(Refer Slide Time: 49:14)



So in this manner you can complete the rest of the table and how many entries stable will have? Q has 4 states; Sigma is 2 symbols, so for each pair of state and symbol I need a line here. So it will have totally 8 lines and this is Delta you can complete this and remember what was q0? The initial state, q0 is the initial state for this machine and this set F consists again only of the state q0, right?

Now before we take up other examples let me formulize a little bit more, I told you what an FSA will define, these FSA is to be, it has 5 components but this business about when to say the string is in the language that was kind of left somewhat (()) (49:51). So let me make that a little more formal. Let me specify it in a manner which is totally unambiguous.

(Refer Slide Time: 50:09)

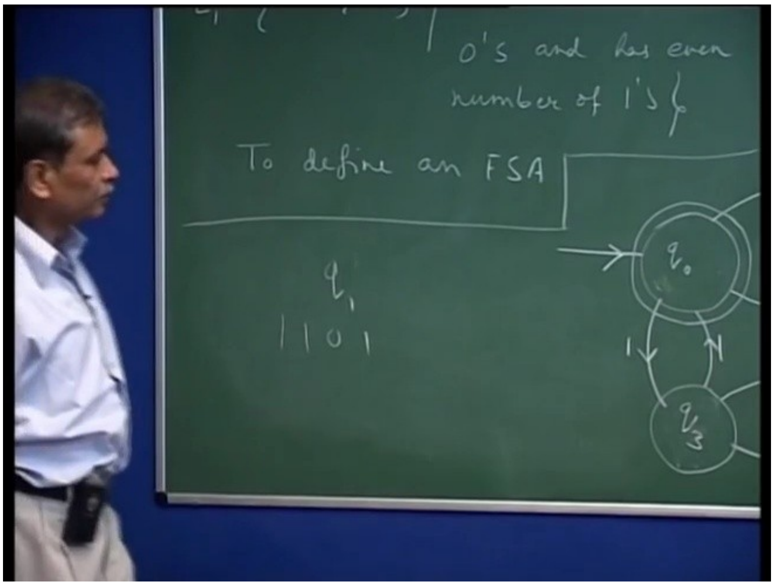


So remember what Delta was? Delta was a mapping from Q cross Sigma to Q but you see actually I can extend this mapping to define one which I if mean some state and then a string comes instead of a symbol, after scanning that string from that state starting from that state which state I shall be.

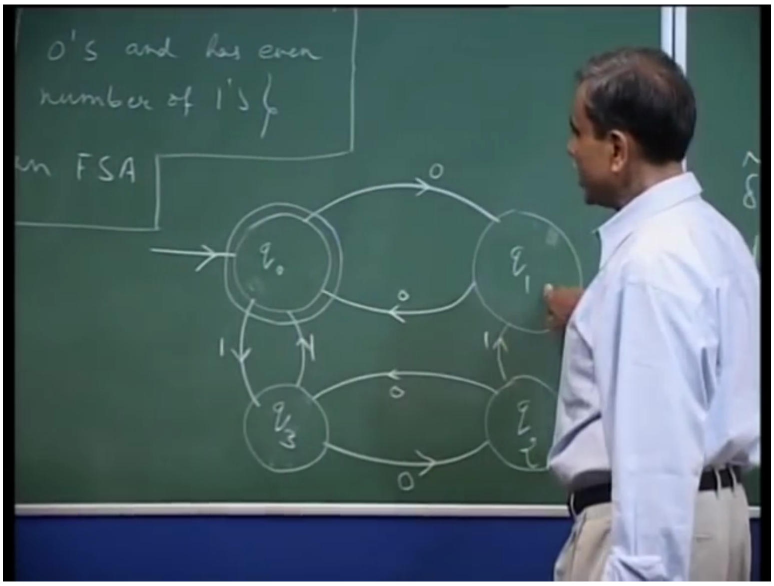
So let me put it this way Delta hat is the function which tells the state, the machine, the automaton will be after scanning the string some string x, starting from the state Q. Formerly what is Delta hat? What is the domain and range? Delta hat is a function which starting from a some state, if you give a string, now what is the notation for any string or set of all strings finite strings?

Of course these strings are over the alphabet Sigma, so this is, so this means it and the string, so I have given a state I have given a string and what is the state the automaton will be? So Delta hat the domain is state string pair and the range is going to be a state.

(Refer Slide Time: 52:53)

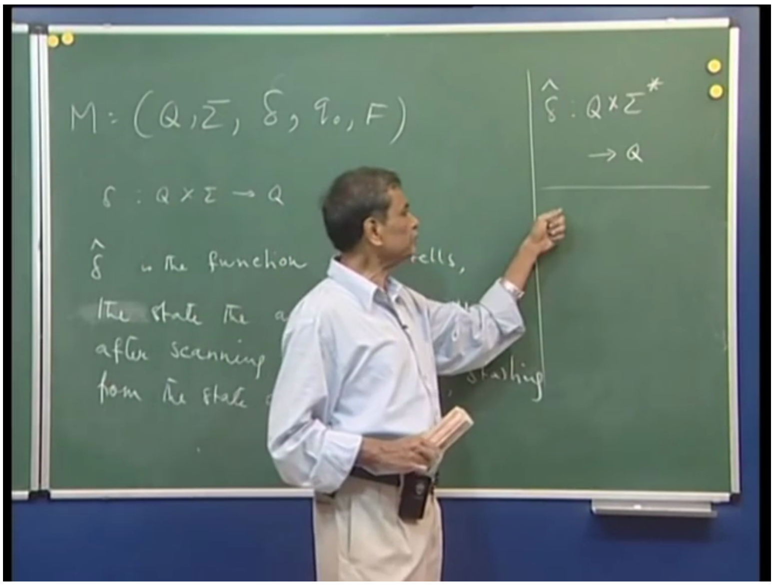


(Refer Slide Time: 53:04)



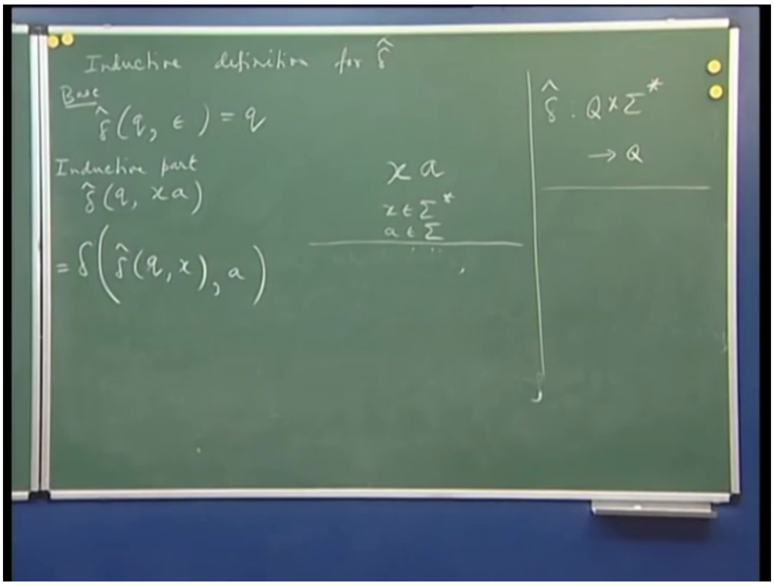
So for example in this supposing I am in Q1 and the string that came is 1 101, this is the state am going to be in? Oh, I did not, right. So q1 then 1 came so I am here then another 1 came I am here then 0 came, remember I will take this edge then 1 came then I am here. So actually this picture tells me even that information starting from any state given any string when the machine or where the automaton will be after scanning that string starting from this state, right?

(Refer Slide Time: 53:43)



You can see this and you can do it for any string but can we do it formally? Can we define this function Delta hat given that function Delta and that is not too difficult. So I want to define this function for all strings which is an infinite set, all finite strings over Sigma. So let me , this is an example of an inductive definition.

(Refer Slide Time: 54:26)



So inductive definitions you have seen other examples that we will start from a base case and then say that suppose I have the definition for something, what happens when I get the next bigger one bigger object? So here the objects are strings and what is the smallest string which is Epsilon, right? So can we answer this that suppose I mean the state q and the string is

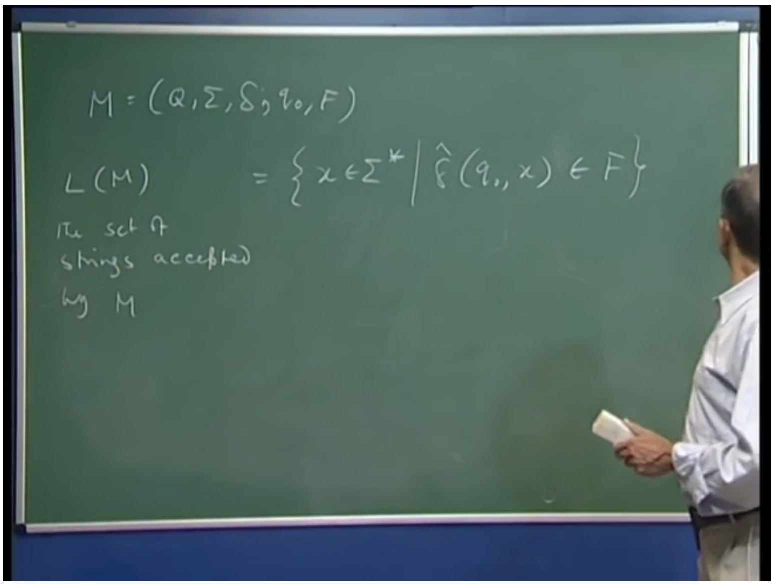
Epsilon, Epsilon is of course has no symbols, so therefore the machine will remain in this state. So therefore it is should be q and now this is the base of the inductive definition and now should be inductive part of the definition.

In case of string I have some string already for which suppose I know the answer and then a symbol came a, so x is a string, a is a symbol, right? So I would like to define Delta hat starting from q, xa what is this? Well, let me say inductively of course I know Delta hat of q on x. So this means what? The state in which the machine will be starting from q and when the string is x? And now symbol has come, another symbol and what is that symbol which is a.

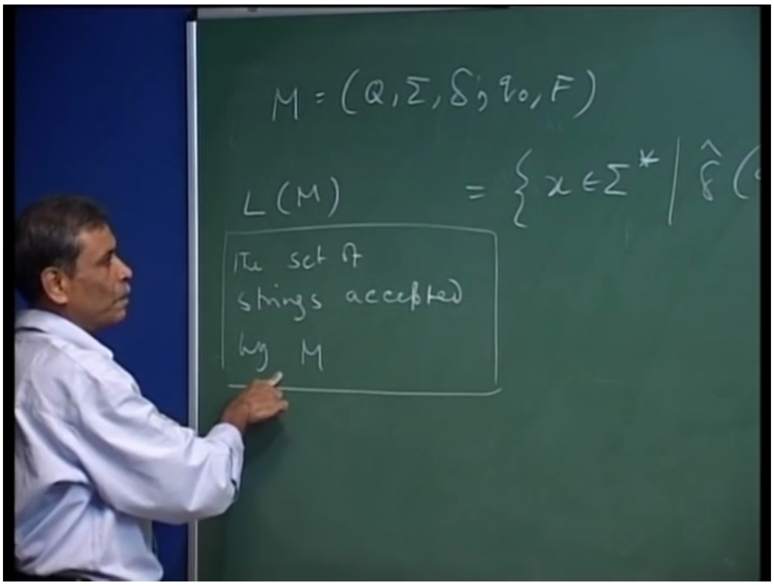
Do you see that this is going to be Delta hat q, xa is equal to Delta of, now this is a state and a, right? This is not difficult to see, what here I am saying is, really what this says that if you know the state, the machine is going to be after seeing the string x then that state when the symbol a came from that state, is a state where that machine will be, is the state where the machine will be on this entire string xa, right?

So this is not difficult but the reason for doing this is that now I have this convenient that I have extended Delta which was, Delta was the mapping from Q cross Sigma to Q and now I have extended it to Q cross Sigma star. Now I do not need this.

(Refer Slide Time: 57:36)



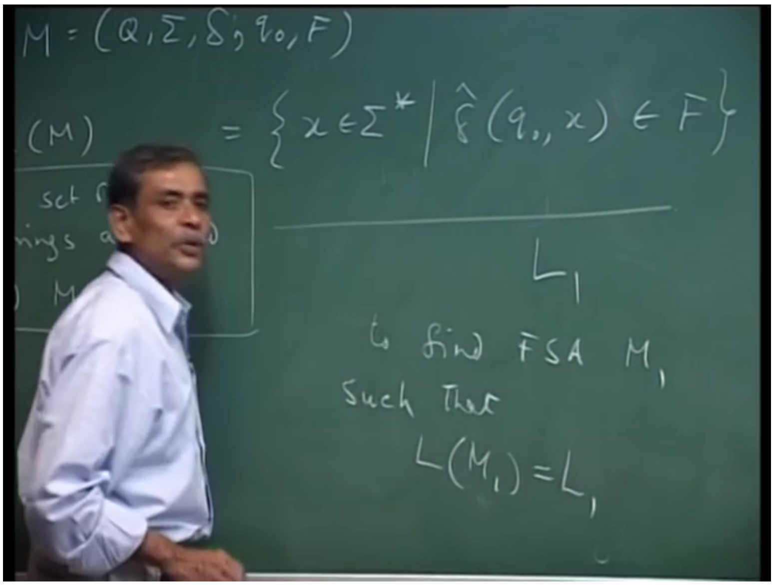
(Refer Slide Time: 59:01)



What I am trying to define is now, that given any automaton like that Q, Sigma, Delta, q0, F, I can define a language which is like this LM to be set of all strings from this alphabet such that Delta hat, q0, x is an element of this F, using this Delta hat we are defining a language which is associated with this machine M. Let me just write it now like this L of L in bracket M which is this and the idea is that this L is the language accepted by this machine M.

So you see given any machine using this I can, using the concept that we have just defined I can associate a language with that machine which is the set of strings accepted by M. Now what we are going to do? From now on for some time when we are discussing this that you will give me a language and my job will be to find a machine M.

(Refer Slide Time: 59:42)



Now let us say you have given me a language L1, so you can see this is a legitimate problem to find FSA finite state automaton M1 such that language accepted by this machine M1, is this language L1 that you give. So in that sense M1 is going to be a description of the language that you have given me. M1 is going to be a very simple algorithm to decide the membership question for the language L1. We will continue this in the next lecture.