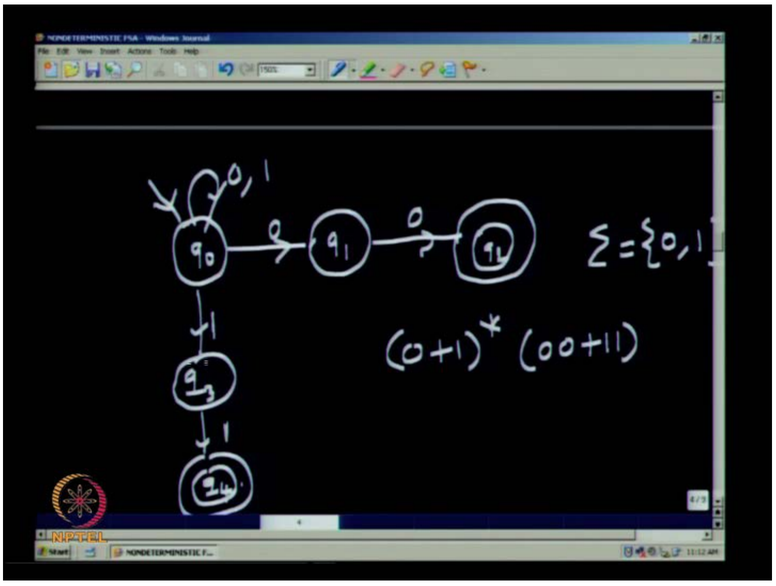
Theory of Computation
Prof. Kamala Krithivasan
Department of Computer Science and Engineering
Indian Institute Of Technology, Madras

Lecture No. # 10
Non Deterministic FSA

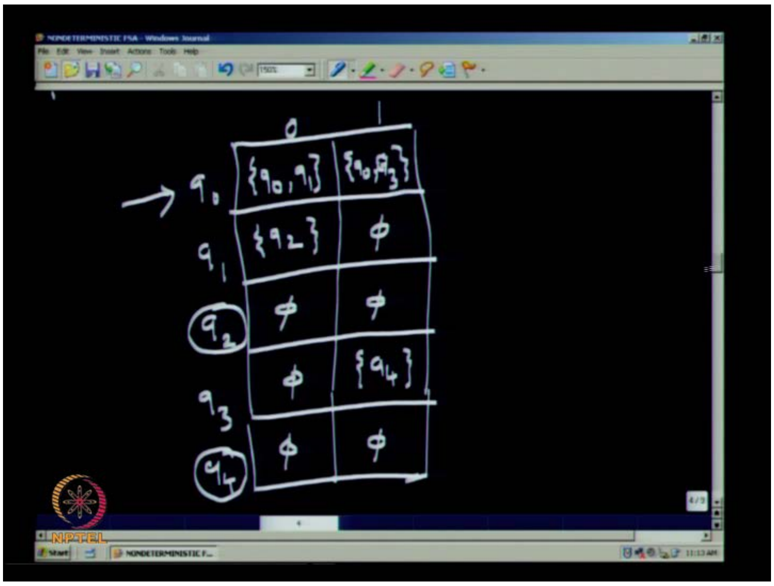
We were considering the conversion of non deterministic FSA into deterministic FSA. One example we consider, we shall consider some more example and see how it works.

(Refer Slide Time: 00:25)



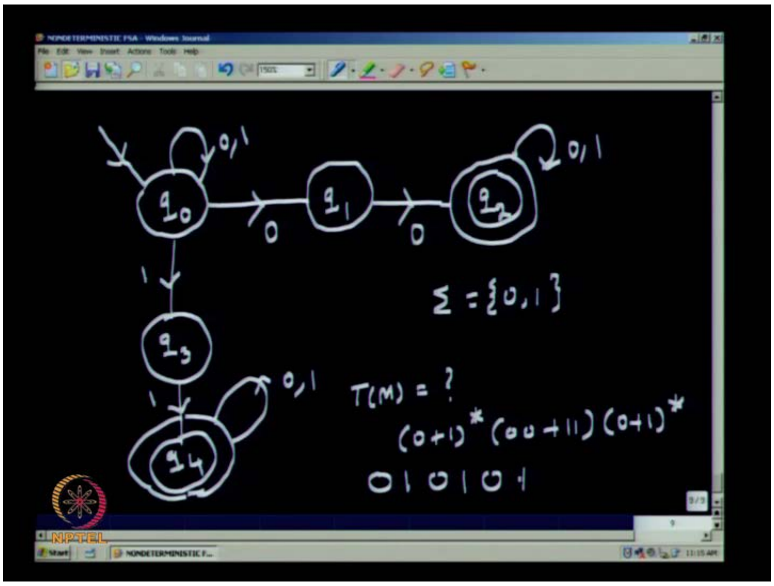
Look at this one. This is a non deterministic diagram, this we consider in the last class itself. The language accepted consists of strings over 0, 1 ending with two 0 (s) or ending with 2 ones. The non deterministic diagram has five states. Why it is non deterministic, because from q naught if you get a 0, there are two possibilities either you can go to q naught or q 1. And if we get a 1 from q naught, you can go to q naught or q 3. So, because of that number of possibilities exist, it is a non deterministic diagram.

(Refer Slide Time: 01:06)



The state table is given by this, q naught is the initial state, and q 2 and q 4 are final states, and within each cell you have a set. Now, how would you convert this into a deterministic diagram? Let us see that.

(Refer Slide Time: 01:56)

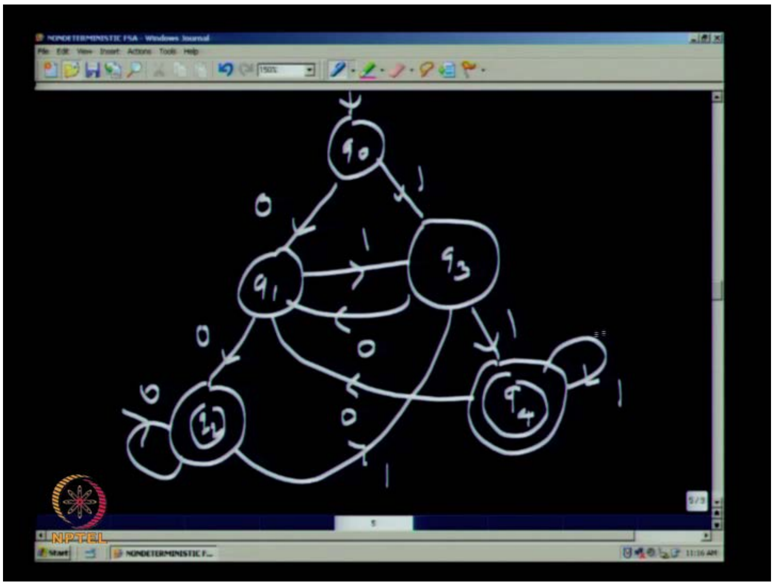


But before that we will consider a similar example of a non deterministic one look at this, it is also a similar diagram, but there is some difference. These portions are the same, but you have an extra loop here. So, what is the language accepted by this machine? The alphabet is 0 1 the language accepted. It consists of I can put it as a regular expression what is this any string of zeroes and ones it should be followed by a 1 0 or 1 1 then you may have any number of zeroes, and ones any string of zeroes and ones this is called a regular expression.

We shall formally define a regular expression in the next 1 or 2 lectures, you can see that the last example which we considered the string ended with the 2 zeroes or ended with 2 ones here it is not necessary should end with a 0 2 zeroes or 2 ones, but in between some where you must have 2 consecutive zeroes or 2 consecutive ones, let us say string of this form 0 1 0 1 0 1 will not be accepted.

Somewhere, you must have 2 zeroes or 2 ones then, after having 2 zeroes or 2 ones you may have any string that does not matter. So, there is a difference between this diagram and the earlier diagram though they are almost similar. Now, one by one let us convert this into deterministic diagram using the subset construction and see what happen. How will the deterministic diagram look like?

(Refer Slide Time: 03:59)

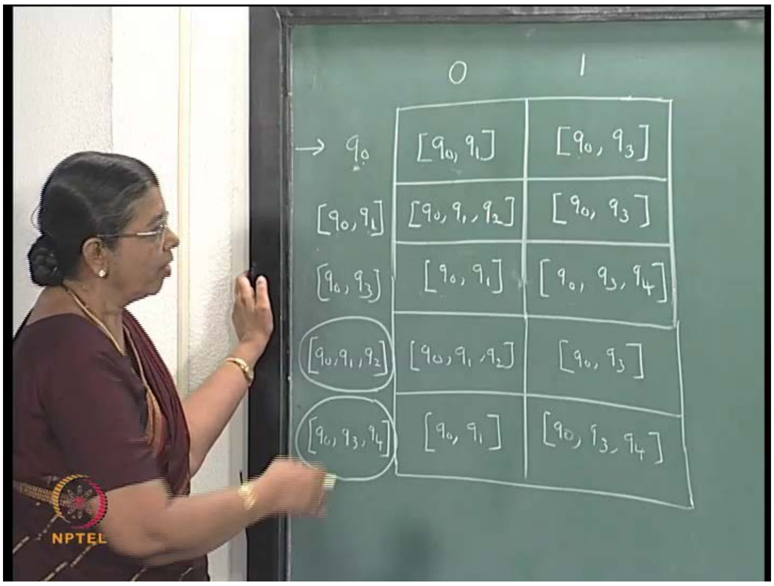


So, let us take the first one which we considered. This strings ending with 2 zeroes or 2 ones the state table is this and I also showed you a deterministic diagram equivalent to that. This is the deterministic FSA from every node there is an arch with label 0 leaving, there is an arc with label 1 leaving and these two are final states this accepts a all strings which end with 2 zeroes or end with 2 one it is a deterministic diagram.

So, try to convert this into deterministic diagram use the subset construction. Now, proceed take 2 or 3 minutes and try to finish the example convert this into deterministic diagram. So, how many states do we get in the deterministic diagram?

The state table for the non deterministic diagram is like this in the deterministic diagram you start with q naught. So, q naught will be the initial state and from q naught you have to fill in this table after getting a 1 what state you will go to that is you have to find a union of this you have to find this will be taken as a state, q naught q 1 will be taken as a single state and q naught q 3 will be taken as a single state.

(Refer Slide Time: 05:29)



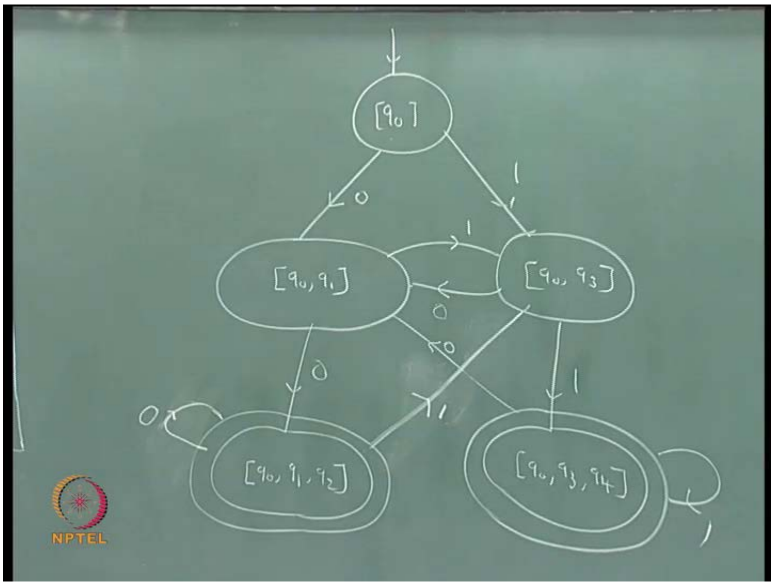
So, you get from q naught, if you get a 0 you go to this state and if you get a 1 you go to this then, from this you have to see what are the states to which you go after reading a 1 or a 1, so from q naught and q 1, if you get a 0 what state you will go to. So, that is from q naught what is the state you will go to, from q 1 what are the states you will go to and then add them find the union this is the definition we have seen.

So, from q naught and q 1, if you read a 1 you will get q naught q 1 and q 2. So, the union of these three is a next state that is a q naught, q 1, q 2 and when you get a 1 what will the next state, you have to find the union of these two that is q naught and q 3 itself in a similar manner. When you find from q naught and q 3, what would be the next state you will go to. If you get a 1 that is a union you have to find it will give you this then it will give you this and so on.

So, continuing like this then, you have to consider this subset as a state and if we get a 1 what are the states to which you will go you will get this. Then from this after getting a 1, what are the states you will go to that you will get this similarly for this. Now, in the original diagram I have seen that q 2 and q 4 is final states.

So, any subset containing q 2 and q 4 will be a final state. So, these two are final states now, this table please note that every state is a subset, but this is single state remember that is single state and it is represented as a subset of the original non deterministic automata.

(Refer Slide Time: 07:42)

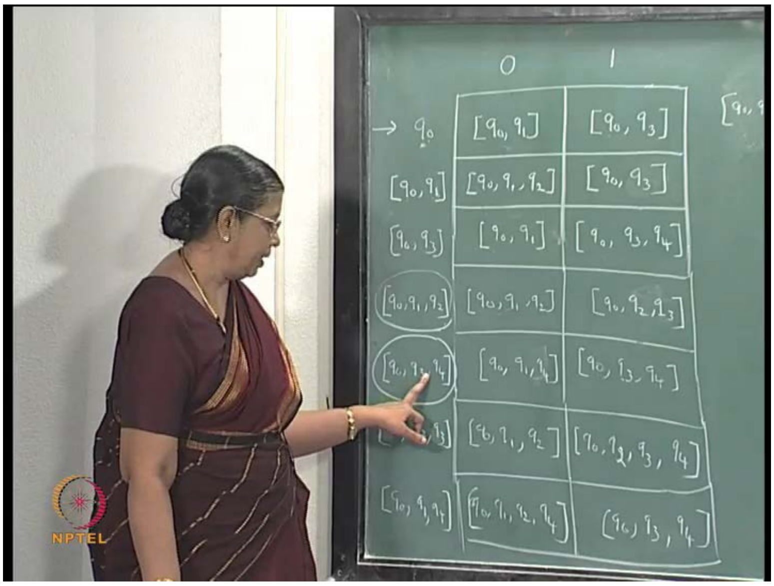


Now, for this table if you draw the state diagram it will be like this. Now, you can see that starting from q naught if you get 2 zeroes you will end up I am sorry the diagram is not correct from q naught q 1 q 2 if you get a 1 you will be here, here if you get a 1 you will be here if you get a 1 you go back here if you get a 1 you go here.

So, this is the state diagram of that that table and it has five states to accept such a language you require at least five states you cannot do it less than five state that we shall see later, but earlier we constructed the deterministic diagram for this without going through the subset construction to the deterministic diagram. You can see that this is exactly similar to that except that in between the I mean the labels are subsets the diagram is exactly the same is it not, this graph if you look at it as a graph this graph is isomorphic to that graph.

So, we have converted this finite state automaton in non deterministic finite state automaton into deterministic automata. Now, do the same thing for this try to construct the deterministic automaton for this, this a slightly different one see how it works. Actually in the deterministic diagram you may have up to 2 power n states isn't it if the non deterministic diagram consists of n states the deterministic diagram can have up to 2 power n states, but mostly it will be slightly more than the slight non deterministic one it will not be 2 power n. We will see the worst case scenario, when it occurs in a moment from q naught q 3, if you get a 1 you go to q naught q 1, if you get a 1 you go to q naught q 3 q 4.

(Refer Slide Time: 10:33)



So, up to this point the same from q naught q 1 q 2, if you get a 1 you go to q naught q 1 q 2, if you get a 1 you go to q naught q 3, but q 2 also will be there, here you will also have q 2 q 3.

So, next you consider you have to also consider that q naught q 2 q 3 let us proceed and see then q naught q 3 q 4, if we get a 1 q naught q 3 q 4, if you get a 1 you get q naught q 1 and q 4, if you get a 1 you go to q 4. So, you will also have q 4 here and from q naught q 3 q 4, if you get a 1 you go to q naught q 3 q 4.

So, this one you have already considered this one you have considered. So, you have to consider this one, q naught q 2 q 3 and also q naught q 1 q 4. So, if you have q naught0 q

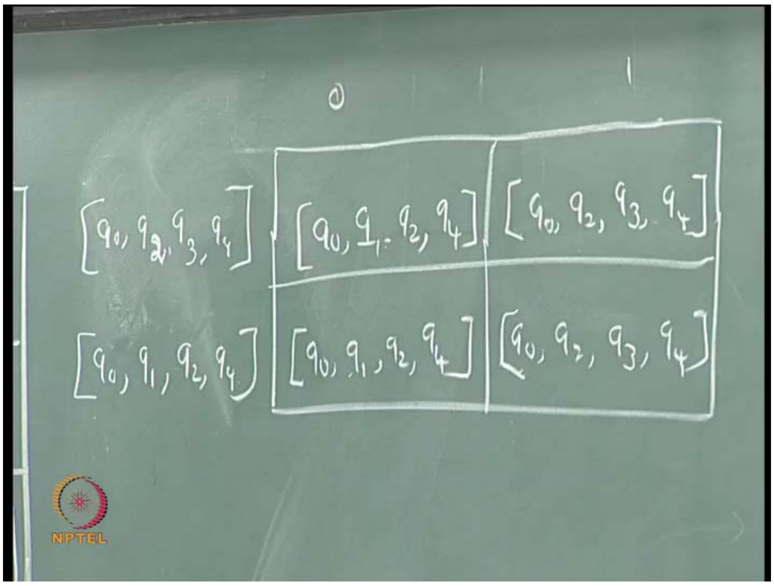
2 q 3, if you get a 1 what we get from q naught, if you get a 1 you can go to q naught r q 1 q naught r q 1 from q 2, if you get a 1 you can get q 2 from q 3 you cannot get a 1.

Now, from q naught q 2 q 3, if we get a 1 what is the next state from q naught, if you get a 1 you can go to q naught q 3 from q 2, if you get a 1 you can go to q 2. So, it will be q 2 q 3 and from q 3, if we get a 1 we can get to q 4 one more state we are getting right. So, you have to consider with this on the left hand side q naught, q 3, q naught q 2, q 3, q 4.

Now, from q naught q 1 and q 4, if you get a 1 what will be the what are the you have to find the union from, q naught if you get a 1 you go to q naught r q 1 from q 1, if you get a 1 you go to q 2 from q 4, if you get a 1 you go to q 4. So, you will get q naught, q 1, q 2, q 4, you have to find a union right. Now, from q naught q 1 and q 40, if we get a 1 what will you go to, from q naught if we get a 1 you go to q naught and q 3, from q 1 you cannot get a 1 from q 4, if we get a 1 you go to q 4.

So, you have q naught, q 3, q 4, which you have already considered. So, this you have already considered this also you have considered this q naught. So, we have to consider the next stage for these two subsets or the states represented by these two subsets q naught, q 2, q 3, q 4, q naught q 1 q 2, q naught q 1 q 2 q 4. Now, what will be the next states in these cases, from q naught if you get a 1 you can go to q naught or q 1 from q 2, if you get a 1 you can go to q 2.

(Refer Slide Time: 16:41)



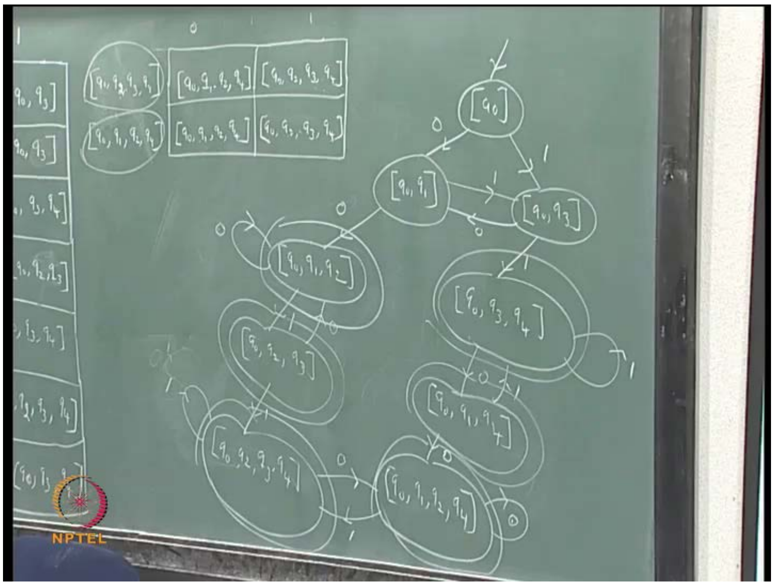
So, this will have q naught, q 1, q 2, from q 3, you cannot get a 1 from q 4, if you get a 1 you go to q 4. So, this will be the now, from q naught q 2, q 3, q 4, if we get a 1 word equal to from q naught if you get a 1 equal to q 2 q naught or q 3 from q 2 if we get a 1 you go to q 2 from q 3 if we get 1 you go to q 4 and from q 3 if we get a 1 you go to q 4. So, you get the same state. So, this will be q naught, q 2, q 3, q 4 then, this one from q naught q 1 q 2 q 4 if we get a 1 you can go to q naught q 1 q 2 and also q 4, but not q 3.

So, here you will get q naught q 1 q 2 q 4 and if you get a 1 from q naught q 3 and q 4. If you get a 1 you can go to q naught, q 3, q 4 the q 3 is not there from q naught if you get a 1 you can go to q naught or q 3 from q 4 if you get a 1 then you can go to 4. So, q naught, q 3, q 4 are there from q 1 you cannot get a 1, but from q 2 if you get a 1 you go to q 2.

So, you will get q naught q 2, q 3, q 4 now, this you have already considered this you have considered. So, no more states subset needs to be considered. So, how many states you are getting 1, 2, 3, 4, 5, 6, 7, 8, 9 states you are getting and this is the initial state the one corresponding to q naught is a initial state what are the final states? The final states whenever there is a final state in the subset that state will be a final state.

So, here the final states q 2 and q 4 are final states in the original diagram. So, you will have q 2, q 4 this is also a final state this is also a final state, this is a final state this is the final state. Now, if we draw the state diagram of this machine it will be like this start from q naught.

(Refer Slide Time: 19:51)



So, if it is a deterministic diagram there should be an arc with label 0 leaving, there should be an arc with label 1 leaving. So, when you get a 1 you go to q naught, q 1 when you get a 1 you go to q naught q 3. Now, from this if we get a 1 you go to q naught q 1 q 2 q naught q 1 q 2, if we get a 1 you go to q naught q 3 from q naught q 3, if we get a 1 you go to q naught q 1 from q naught q 3, if you get a 1 you go to this.

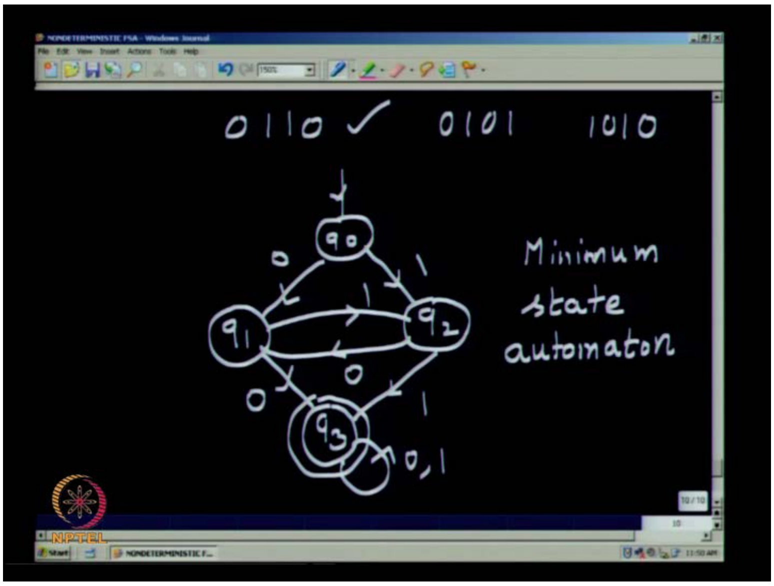
So, q naught, q 3, q 4 now, from this there are two possibilities from q naught q 1 q 2, if we get a 0 you go to the same state if you get a 1 you go to q naught, q 2, q 3 another state q naught, q 2, q 3 from q naught, q 3, q 4. If you get a 1 you go to q naught, q 1, q 4 another state q naught q 1, q 4 (( )) get a 1 this I have not this 1 from q naught, q 3, q 4. If you get a 1 you go to q naught, q 1, q 4, if we get a 1 you go to q naught q 3 q 4 that is the self loop will be there .

So, we have drawn seven states now, two more states you have to draw from q naught, q 2, q 3, if we get a 1 you go to q naught, q 1, q 2. If you get a 1 you go to q naught 0, q 2, q 3, q 4, q naught, q 2, q 3, q 4, from q naught, q 1, q 4. If you get a 1 you go this we have drawn. This portion we have not drawn q naught, q 1, q 4, if we get a 1 you go to q naught, q 3, q 4 say this will be from q naught, q 1, q 4 if you get a 1 you go to q naught, q 1, q 2, q 4 another state q naught, q 1, q 2, q 4 (( )) this is when you get a 1 when you get a when you get a 1.

Now, these two you have to represent here from q naught, q 2, q 3, q 4 that this state if you get a 1 you go to q naught, q 1, q 2, q 4, if you get a 1 you go to q naught, q 2, q 3, q 4 the same state from q naught, q 1, q 2, q 4, if you get a 1 you go to the same state, but if we get a 1 you go to q naught, q 2, q 3, q 4.

So, the initial state we have marked, what are the final states? Any subset containing q 2 or q 4 will be a final state. So, this is a final state this is a final state and this is a final state again all these are final states. So, almost similar diagrams they are not same they are similar, but the subset construction in one how many states you had you had five states actually, from five states if you construct a deterministic automaton you may go up to two power five, three, two states, but in one you are just getting five you need not have to go beyond that in another one you are getting nine states using the subset construction.

(Refer Slide Time: 26:35)



So, this we see, but is this is the only diagram you can have see what is a language accepted the language accepted has either 2 consecutive zeroes or 2 consecutive ones it can have consecutive zeroes consecutive ones everything, but if it is a string of the form 0 1 0 1 1 0 1 1 0 will be accepted 0 1 0 1 such a string will not be accepted 1 0 1 0 will not be accepted. You can see in this diagram that you get only after getting 2 ones only you go to your final state or after getting 2 zeroes only you go to your final state then it will be accepted.

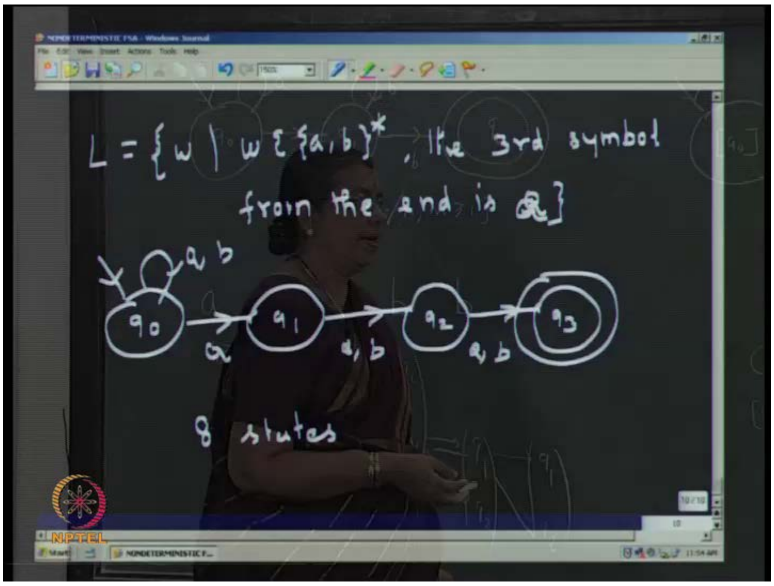
Now, look at this diagram (no audio 26:58 to 27:46) is this deterministic diagram or a non deterministic diagram. What is the condition for the deterministic diagram from every node there should be one arc with label 0 leaving and one arc with label 1 leaving, see whether it is true and you cannot have two arcs with label 0 leaving, from the same node or 2 logs with label 1 leaving, the same node q naught there is one arc with label 0 1 arc with label 1 from q 1 there is one arc with label 0 one arc with label 1 from q 2 there is one arc with label 0 one arc with label 1 from q 3, there are I mean this is supposed to be 2 arcs one with label 0 and one with label 1. What is the language accepted by this the language accepted is the same see as long as you do not get 2 ones or 2 zeroes you will be within this 0 1 0 1 0 1 you will be here or if you started with the 1 you will get 1 0 1 0 1 0 like that, but once you get 2 consecutive ones or 2 consecutive zeroes you will go to q 3, then afterwards it is immaterial what string you get it will be accepted isn’t it.

So, this deterministic diagram accepts the same language and what we have seen there that is also a deterministic diagram it also accepts the same language it does not accept anything more it accept the same language that has got nine states whereas, this has got only four states.

So, what you can achieve with nine states you are able to achieve with four states, this is called the minimum state automaton. So, the minimum state automaton for accepting this language has got four states, there can be other automata nine states, eleven states you can have many more also which will accept the same language. So, the earlier example which we considered the ending with 2 zeroes or ending with 2 ones. This is the deterministic diagram, but the minimum state automaton is also this you cannot go beyond or you cannot go less than five states you need you will require at least five states for this. How to construct the minimal minimum state automaton we shall see in the next few lectures.

Now, the worst case scenario see as I told you if you have n states. If you have n states then the worst case you may have up to 2 power n states, but generally you will find that you need not go up to that level what sort of language will have this.

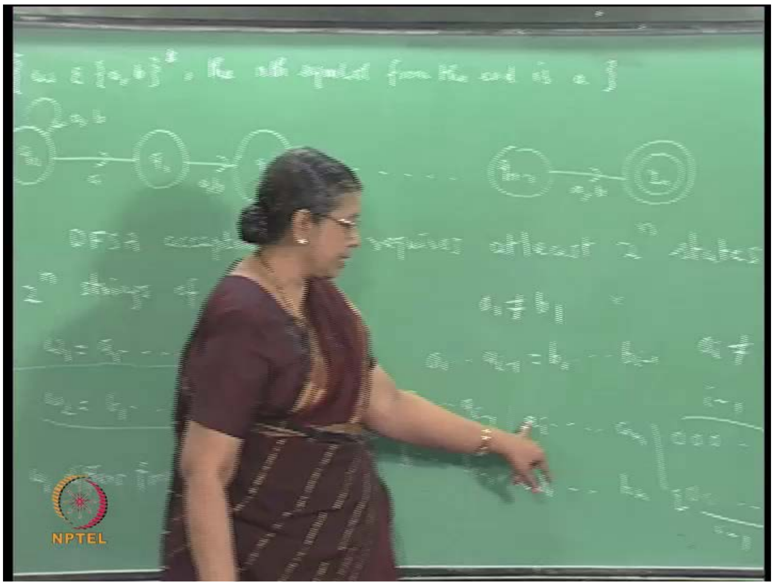
(Refer Slide Time: 31:34)



Suppose I construct a language L, L consists of strings w where w belongs to a comma b star string of a comma string of a(s) and b (s), the third symbol from the end is 0 from the last the third symbol is a 1. Suppose I say like this the diagram non deterministic diagram you can very easily draw, you can draw a non deterministic diagram like this make sure that you can have any I mean sorry I used a b and then I use 0 1 sorry this is a b a is a this is a.

So, this is a b this a b. So, this make sure that you can read any a b then, one a the last two symbols can be anything third symbol, from the end is a this is a non deterministic diagram and when you try to convert this into deterministic diagram you will find that you will require a eight states.

(Refer Slide Time: 33:52)



Now, we have seen that the worst case scenario occurs. When we consider a language something like this the language L w belongs to a b star the nth symbol from the end is a, you have a signs of a (s) and b (s) but there a nth symbol from the end is a and it cannot be a b.

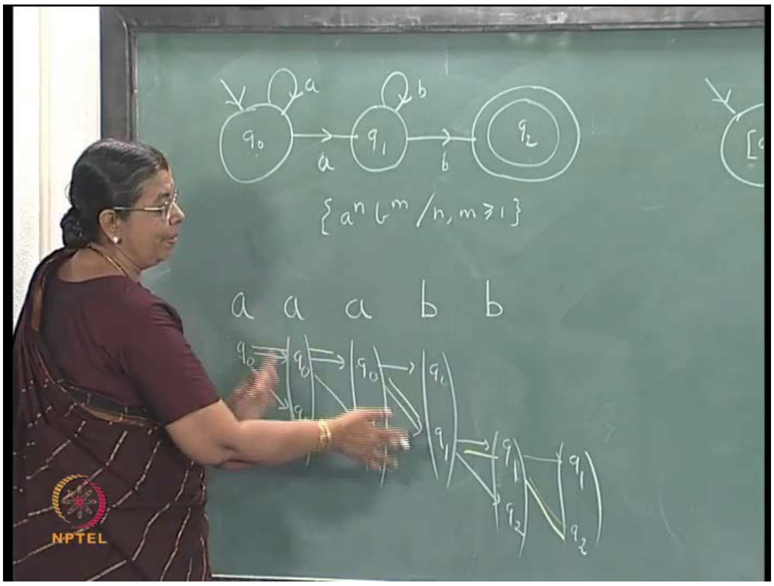
So, the non deterministic diagram you can very easily draw for this it will be like this you can have any string of a (s) and b (s) then a q 1 q 2 a b q n minus 1 a b q n you can read any a and b and then, once you read a afterwards you can read only n minus 1 symbols which can be again a b all possibilities will be there, to accept this as a by a deterministic automaton the d FSA accepting this requires at least 2 power n states. Why, there are 2 power n strings of length n using a and b you can have 2 power n strings of length a n suppose w 1 is 1 string a 1 a 2 a n w 2 is a b 1 b 2 b n.

Now, starting from q naught after reading this and after reading this you have to be in 2 different states you cannot be in the same states. So, for 2 power n strings you require 2 power n states. What happens if they are in the same state after reading two of them suppose starting from q naught after reading this you are in state q after reading this also you are in state q what happens for two different strings. The two different strings means some symbols they have to differ w 1 differs from w, w 2.

Suppose the first symbol is different if a 1 is not equal to b 1 then, this string begins with the a this string begins with the b, this has to be accepted this has to be rejected. So, q has to act both as a final and the non final state and that is not possible. So, in this case it is not possible suppose a 1, a 2, I a I minus 1 equal to b 1, b 2, b I minus 1, but a I is not equal to b I then what happens starting from q naught after reading a 1 to a I minus 1 which is the same as b 1 to b I minus 1 it is in some state r say then, after reading a I to a n, b I to b n it is in q then consider it is followed by I minus 1 zeroes.

Now, starting from q after reading I minus 1 zeroes it goes to a state p say. The sequence of I minus zeroes can be taken as I minus 1 a (s) in both cases and I minus 1 b (s) in both cases. So, you find that this string the nth symbol is from the last is a in this string the nth symbol from the last it is a b. One has to be accepted the other has to be rejected. That means, p has to act as a final state as well as a non final state which is not possible.

(Refer Slide Time: 38:49)

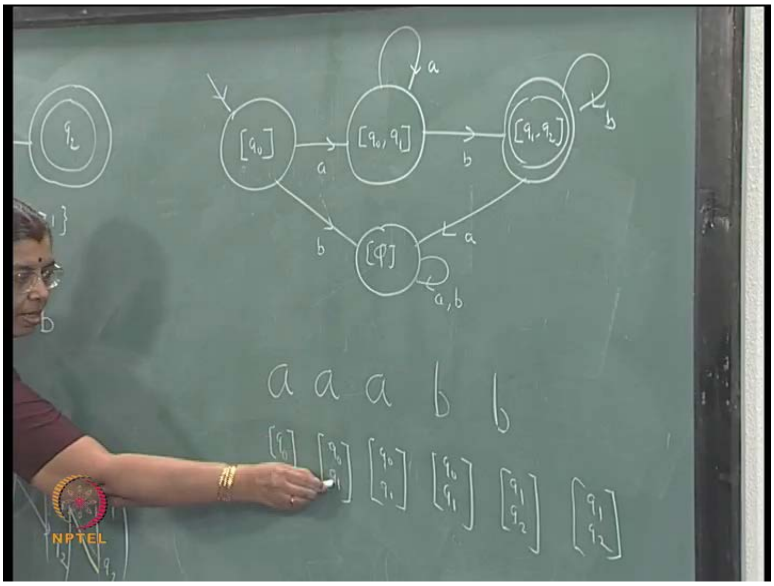


So, you find that any deterministic final state automaton accepting such a language will require at least 2 power n states. So, we have seen this diagram q naught, q 1, q 2, a this is a non deterministic diagram a, which accepts a language a power n b power m n, m greater than or equal to 1. And when we constructed a deterministic diagram from this the diagram was like this from q naught after reading I a the possibilities are you can go to q naught or q 1 from q naught and q 1, if you get a b q 1, q 2 this is the dead state.

So, using the subset construction from this diagram, we obtained this diagram let us take a string a a, a, b, b, and see how it is accepted here and how it is accepted there. So, start from q naught after reading a, a ,you are reading you will be reading a 1 in q naught you are reading this then, after reading a from q naught there are two possibilities we can go to q naught or q 1 again in q naught if you read a, a, there are two possibilities we can go to q naught or q 1 from q 1 you cannot read a a.

So, this sequence stops again from q naught if you read a, a, you can go to q naught or q 1 from q 1 you cannot read a, a, from q naught you cannot read a b from q 1, if you read a b you can go to q 1 or q 2 again I am sorry from q 1 if you read a b you can go to q 1 or q 2 from q 1, q 2 you cannot read anything from q 1 again if you read a b can go to q 1 or q 2, if one of them is a final state the string will be accepted this sequence of states this sequence of states take you to q 2 and the string will be accepted.

(Refer Slide Time: 43:10)



Now, how is this if you look at it as a subset like this each one of them correspond to a state there now if you look at the same string a, a, a, b, b, you start from q naught after reading a, a, you go to the state q 1, q naught, q 1, and from this you get a, a, you go to q naught, q 1 from this state after reading a, a, go to again q naught, q 1 from q naught, q 1 after reading a b you go to q 1, q 2 and from q 1, q 2 after reading a b you go to q 1, q 2 note the correspondence between this and this see if here you have q naught. Then, q naught q 1, q naught q 1 these are the possibilities in the non deterministic here. It is a single state q naught then after a, a, you have q naught, q 1 here the subset the state corresponding to a subset q naught, q 1 then q naught, q 1, q naught, q 1 etcetera q naught, q 1, q 1, q 2, q 1, q 2, q naught, q 1, q 1, q 2, q 1, q 2.

So, you see the correspondence between that what we have considered earlier only we are considering here, but to explain the correspondence between many things are accepted we consider this example any other example which we have considered. So, far we can again take and consider.