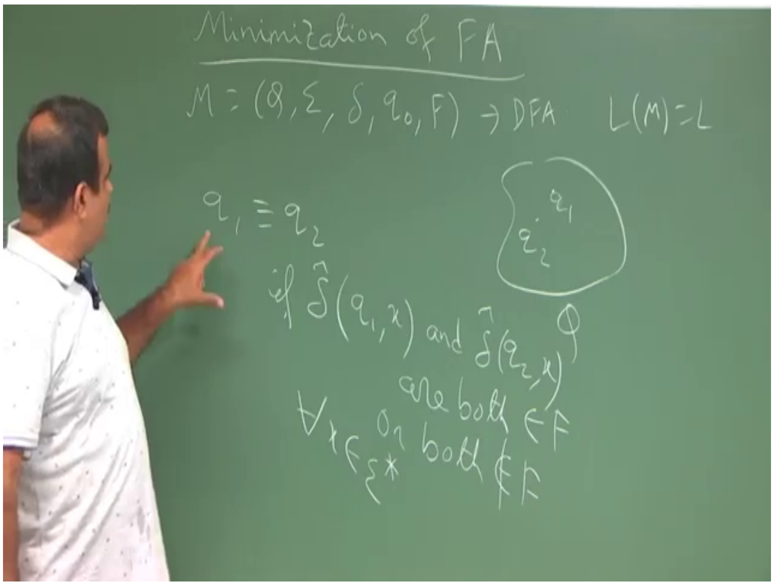
Introduction to Automata, Languages and computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 31
Minimization of FA

(Refer Slide Time: 00:18)

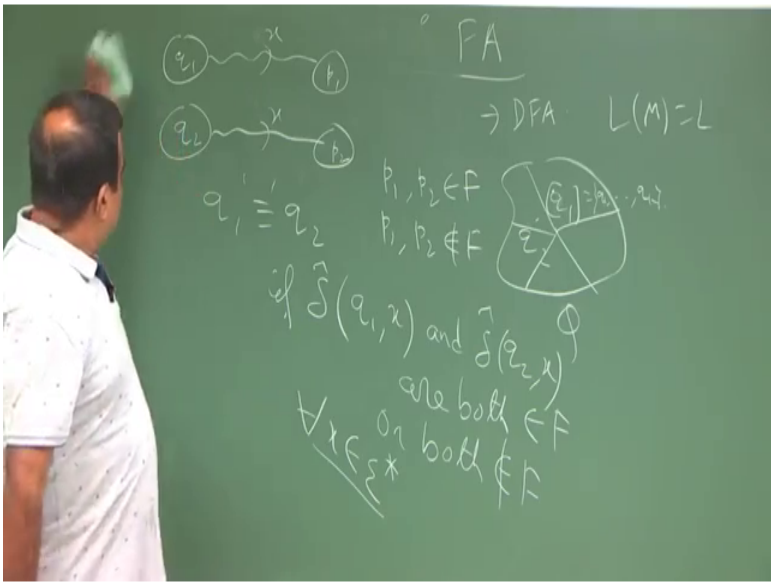


So, good day everybody, so, we will talk about the minimization of the finite automata. So, given a automata, given a regular set we have corresponding to the automata DFA. So, the question is that DFA may have many states which may not be required to accept that language. So, we want to reduce the number of states, we have to minimize the DFA, like that is the minimization process we will learn. So, given a DFA which is again a (Refer Time: 00:52) like this, q 0 F, this is the DFA which is accepting the language this.

Now, we want to minimize this DFA that means, we want to reduce the number of states, we want to reduce the number of states, so that we want to construct a new DFA from this DFA, but our language accepting should be same, the it should accept the same language ok.

So, for that what we are going to do? We are going to define the equivalence relationship between two states. So, this is our Q, this is our Q; Q is the set of states which is finite. So, we defined equivalence relation we said say two state, q 1, q 2; q 1, q 2 they are equivalent, they are equivalent if delta head of q 1 x and delta head of q 2 x are both either final state or both are non-final state. So, either both belongs to a final state or both belongs to non-final state. And that is true for all x; x is the string arbitrary length string, x could be any arbitrary length string. If this is satisfied then we say q 1 is equivalent to q 2.

(Refer Slide Time: 02:45)

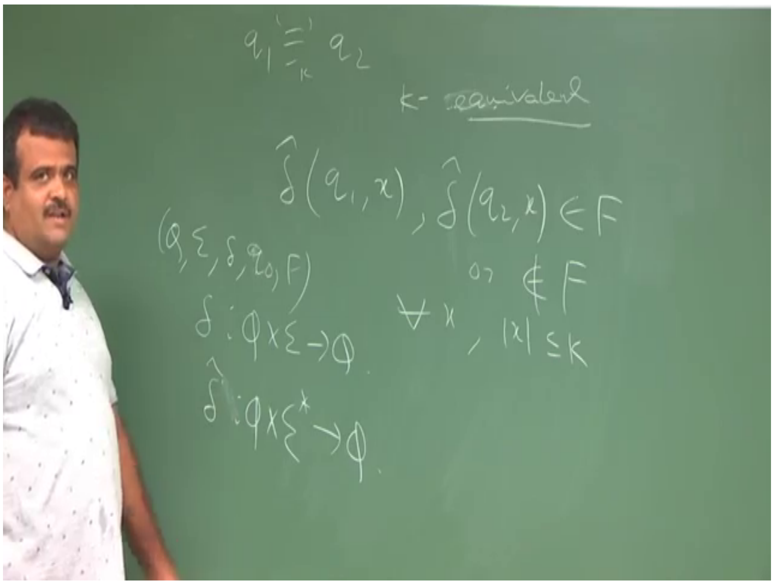


That means, we start with q 1 this is our q 1 we read the string x, if we are reaching to some state say p 1 and we start with q 2 which is the same string if we reaching to p 2, then they then if p 1 p 2 either both in F or p 1 p 2 both is not final state. It is not that if p 1 is belongs to a, p 2 is does not belongs to a (Refer Time: 03:19) like that. So, either we are going to final state or we are going to non-final state, a rejected state. If that is happening then we say the state these two states are equivalent ok.

Now, now we are going to we are this we can show this equivalence relation is a, this relation is a equivalence relation and this will form a partition over the set like over the q. So, it will form a partition like this. So, this is the all set which are equivalent with q, so say q 1, q 2, q k like this. That means, given any string the if q 1 is reaching to some state if it is final state then q 2 will also reach to that same state; so, that is the idea.

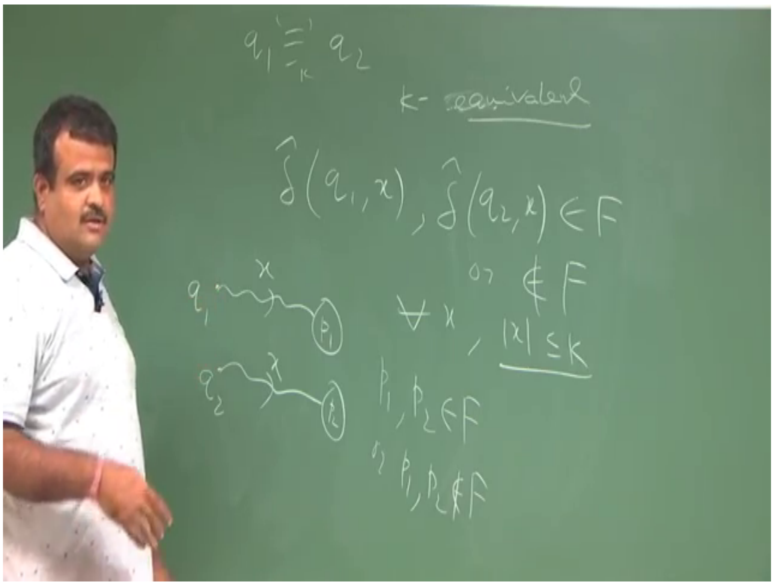
Now, if we have that then what is the why we are trying to mod, I mean how you are trying to find the this classes, these are called equivalence classes; that is one question. Another question is how will find this relationship because this is true for all x, but here x is infinite because it is for all the string of a arbitrary length which is infinite. So, we cannot test for infinite string. So, we can do one thing which is called instead of general definition we can define the k equivalence.

(Refer Slide Time: 05:00)



We can define this, we say q 1 is k equivalence with q 2 this is the k equivalent, k equivalent is q 2 if delta hat of q 1 x and delta hat of q 2 x are reaching to final state or both are reaching to rejected state. And if this is true for all x such that the length of the x is less than equal to k. Delta hat this is the extended transition rule which is over the string. We know the delta, delta is this is Q sigma delta q 0 F. So, delta is function from Q cross sigma to Q, it is the transition rule. Now, using that we can extend this delta hat which is which is taking a string q 0. So, this is recursively defined this, we know this definition.

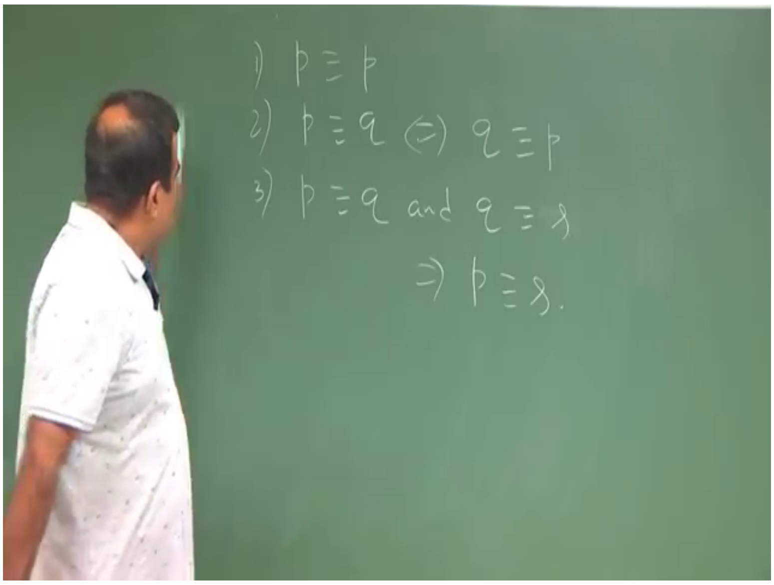
(Refer Slide Time: 06:20)



So, that means, starting from this, starting from q 1 with the string of length at most k. Any string of length at most k if you are going to some state p 1 and starting from q 2 we are keep on reading the string if we are reaching to some state p 2 then there will be k equivalent if p 1 p 2 both belongs to F or both does not belongs to F, both are, if both is going to accept a state or both is going to rejected state then we say these two are k equivalent.

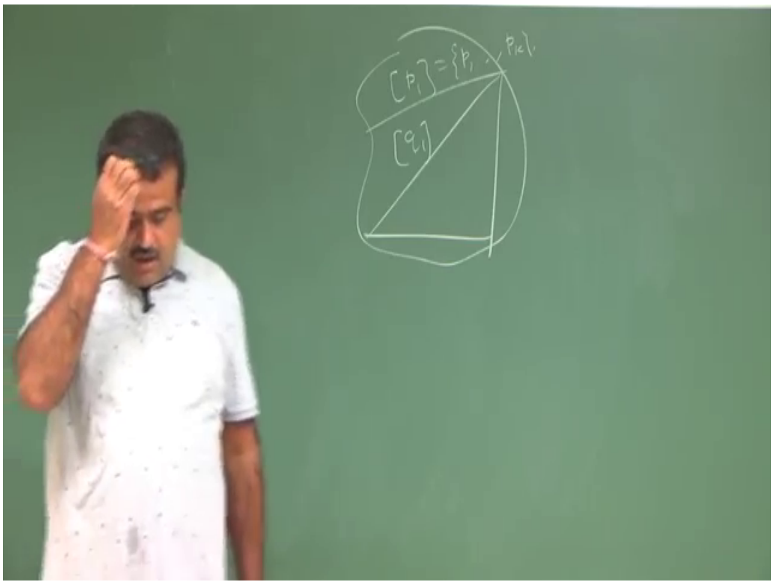
Now, if we have k equivalence then how we can get the k plus 1 equivalence. So, this is a recursive definition of recursive way of constructing the k plus equivalence.

(Refer Slide Time: 07:29)



So, k equivalent this equivalence is a equivalence relation; that means, it is reflexive symmetric transitive because if reflexive means the p is equivalent to p that is obvious, this is reflexive. And if p is equivalent to q this implies q is equivalent to p, this is also obvious. Now, if p is equivalent to q and q is equivalent to r or say s, this implies p is equivalent to r; that means, this is a equivalence relation and we know every equivalence relation form a partitions.

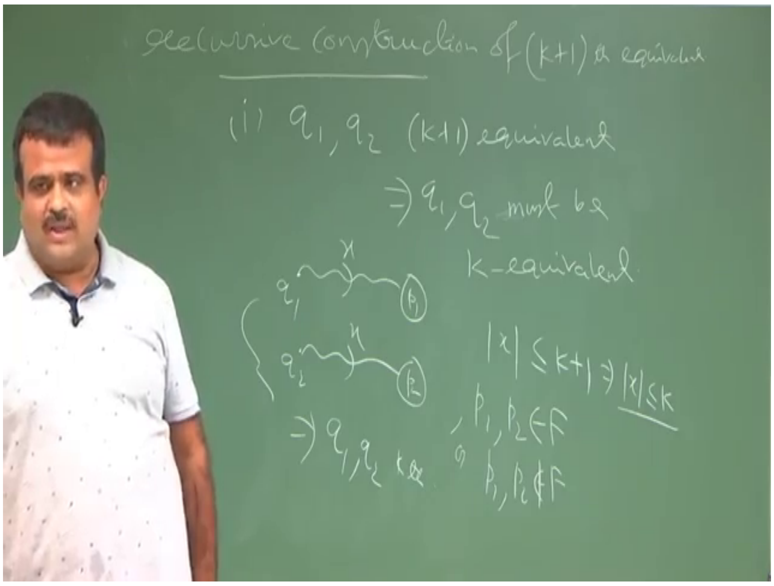
(Refer Slide Time: 08:09)



So, it will form a partition about this set q. So, this is a classes at p 1 class like this, q 1 class like this, suppose p 1 class is having p 1, p 2, p k. So, these are all string these are sorry, these are all states which are k equivalent ok. And if this is true for all k then this is these are called, this we call the equivalent in general, I mean if this is true for all p.

So, then we will merge this will collapse these states which are equivalence. We have a single state in state of this class. So, we will see that why we can do that. So, first let us find out the recursive way how we can find the k equivalence, like k plus 1 equivalence ok.

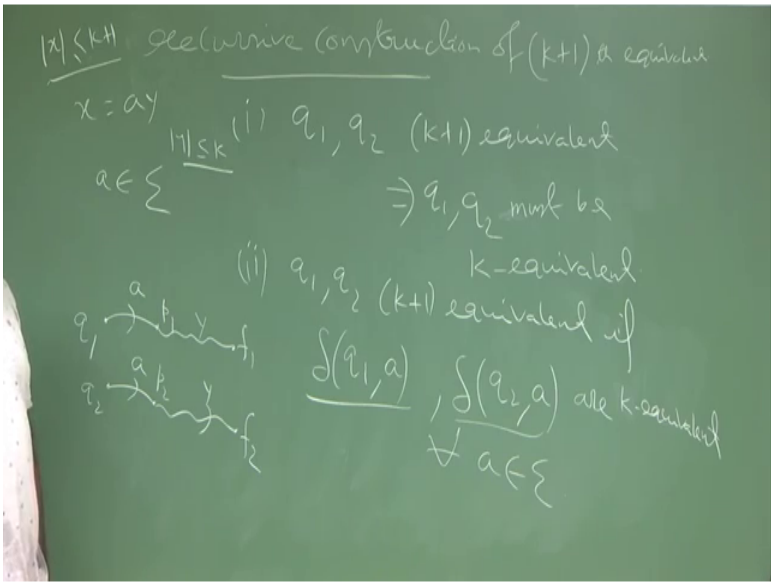
(Refer Slide Time: 09:04)



So, recursive construction of k plus 1 equivalence. Now, for that we need to observe two things, first one is if two state q 1 and q 2 are k plus 1 equivalent then there has to be k equivalent, q 2 must be k equivalence. Why? Because, so, suppose that k plus 1 equivalent, so there k plus 1 equivalent means we start with q r, we read a string x with a length at most k plus 1, any string of length at most k plus 1 and we go to p 2 now p on p 2 will be both are belongs to F or both are does not belongs to F. And this is true for all string of length up to k plus 1.

So that means, this has to be true for all string of length up to k, so that means, this has to be true for all string this imply this has to true for all string less than k. So, that is nothing but kth equivalent or kth equivalent by definition ok. So, k plus 1th equivalent means there has to be kth equivalent. This is the first condition.

(Refer Slide Time: 11:17)



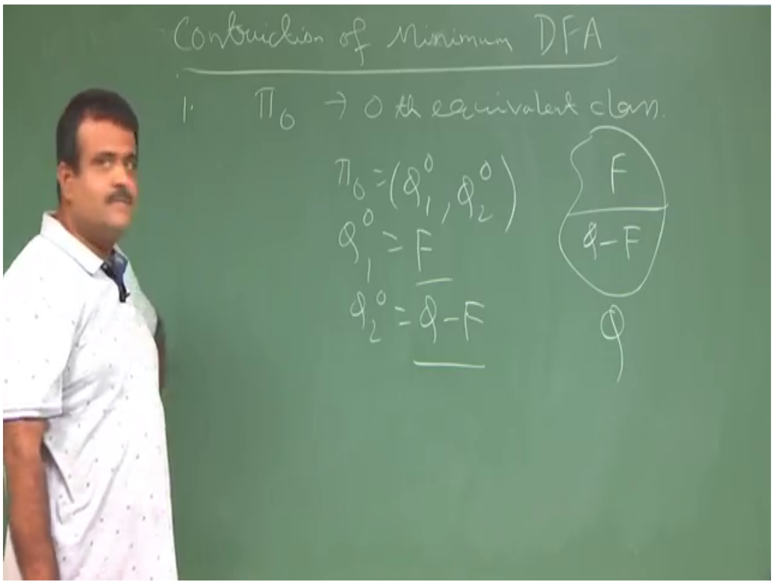
And the second one is; so this will give us the algorithm to construct the equivalent classes, ok. The second one is if, so this one our k plus 1th equivalent if delta of q 1 a, this is one state, and delta of q 2 a are both our k equivalent. And this must be true for all a coming from the alphabet symbol, this must be true for all a are coming from the alphabet symbol.

Why it is true? Because, so we are looking for k plus 1th equivalence. So, we are at q 1 we take a arbitrary alphabet symbol a, we take the move suppose we are going to say p 1, we are going to p 1. Now, from here and from here we can go to with some x which some say, say x if you take a string a y and her length of this is at most k plus 1, length of this. So, length of y is at most k because we have used on one symbol over here.

So, now, with this y we must go to some f 1 and here also with q 2 with a we go to say p 2 and with this y we go to f 2, ok. Now, we are telling these are these two are kth equivalent these two are k plus 1th equivalent; that means, these two should reach to the state which are both accepted or both rejected state. That means what? That means, we already use one symbol so that means, these two must be kth equivalent because if they are not kth equivalent if this is going to accept a state and this is going to reject a state then these two are cannot be k plus 1th equivalent ok.

So, to be these two are kth equivalent if we use one symbol, so remaining strings should reach to the either accepted state or rejected state in the both the input, so both the operations on this. So, that means, there has to be the delta of q 1 a and q 2 delta of q 2 a must be kth equivalent for all a, that is important. So, these will give us a recursive way to construct the equivalence classes ok. This will give us. So, this we can justify.

(Refer Slide Time: 14:38)

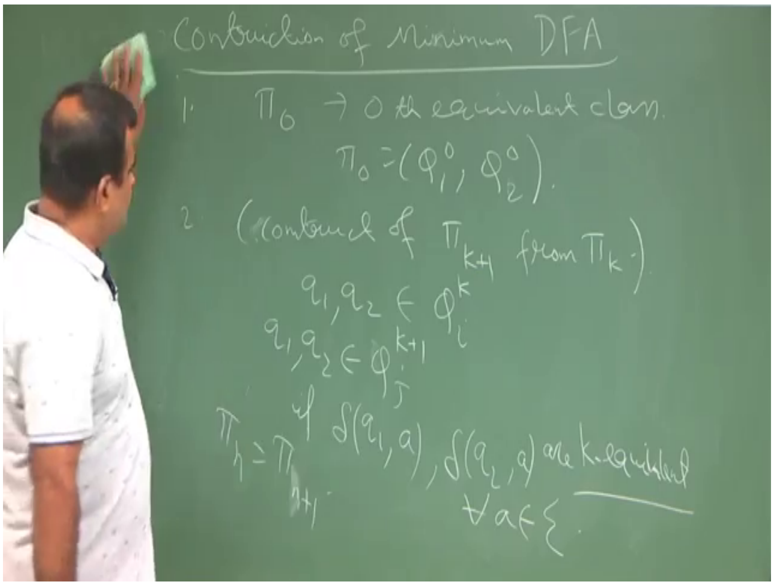


So, this will give us the construction of minimum automata minimum DFA. So, first what we do? First we construct the pi 0, I mean the equivalence classes 0th equivalence class pi 0. So, pi 0 is your what? So, this is our q. So, it will partition into two part. So, pi 0 is 0th equivalent means that we are not reading any string epsilon move, but in the DFA there is no the epsilon move.

So, that means, pi 0 will be two class; one is Q 1 0, another one is Q 2 0, but Q 1 0 is the F set of all accepting string because we if we are acceptor state, if we are at acceptor state then without not reading anything we are remain at accepted state because here we are not allowing any epsilon move.

And if we are at rejected state say some r 1 then with epsilon move we are add rejected state because we do not have epsilon move here. So, that is Q 0 1 and Q. So, this is having two partition Q minus F; F and this is Q minus F. So, only two partition this is, set of all accepted state set of all rejected state ok. Now, from here how we can construct the pi 1.

(Refer Slide Time: 16:47)

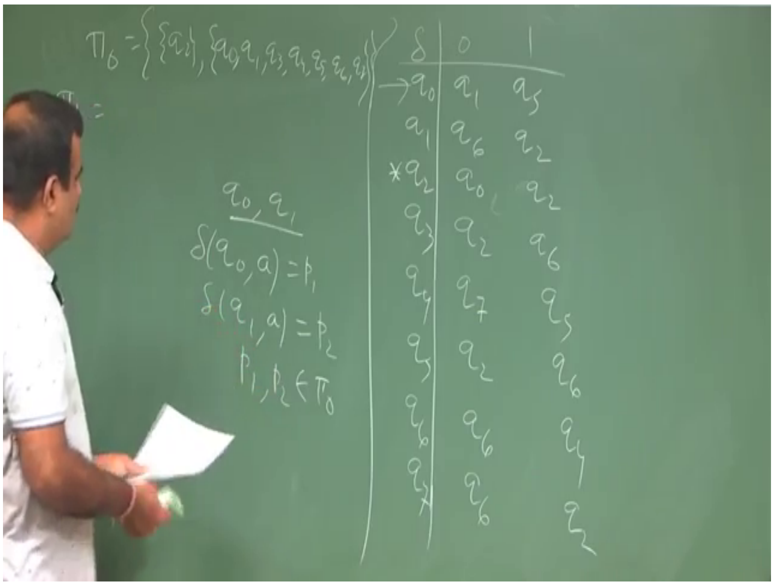


So, pi 0 is this, so, this is the initials ok. Now, the construction of; construction of pi k plus 1 which is the equivalence classes for k plus 1th equivalent from pi k ok. So, we take two state from the one equivalence class of pi k. There are say many equivalence class of pi k, pi k is the equivalence classes of for the kth equivalent. So, one such, so we take a state which are belongs to the same class, so, this is our pi k.

Now, these two will be k plus 1th these two will be in the k plus 1th equivalent if so these two will be some pi J k plus 1 come in k plus 1 equivalence class if we know the definition if delta of q 1 a and delta of q 2 a or k equivalent if they are in the same class. And this must be true for all a, this must be true for all a. If we can do that then this is kth equivalent I mean k plus 1th equivalent ok. So, if you have a kth equivalence class then we can check we can take two state from there and then we can check whether we can partition it again or not. So, we will take an example.

So, this way we continue until it is to a condition that pi n is equal to pi n plus 1. So, there is no further division is possible, so that means, it is converge and that pi n is our equivalence classes in general. So, that we are going to construct through an example. So, this is the idea. So, now, we are going to construct this.

(Refer Slide Time: 19:45)



So, suppose we are given this automata say we have say 8 state and we have two input 0 and 1. So, q 0 this is the starting state which is going to q 1 q 5 and q 1 is going to q 6 q 2 and q 2 is going to q 2 is also a final state q 2 is going to q 0 and q 2 and q 3 with input 0 this is our delta, q 3 with the input 0 it is going to q 2 and with the input 1 it is going to q 6. And from q 4 we are getting q 7 and q 5 and from q 5 we are getting q 2 and q 6 and from q 6 we are getting q 6 as well as q 4. Last one q 7 have 8 state, so q 6 q 2. So, suppose this is our given automata. So, these are the states and this is the these are the transition for the inputs 0 and 1.

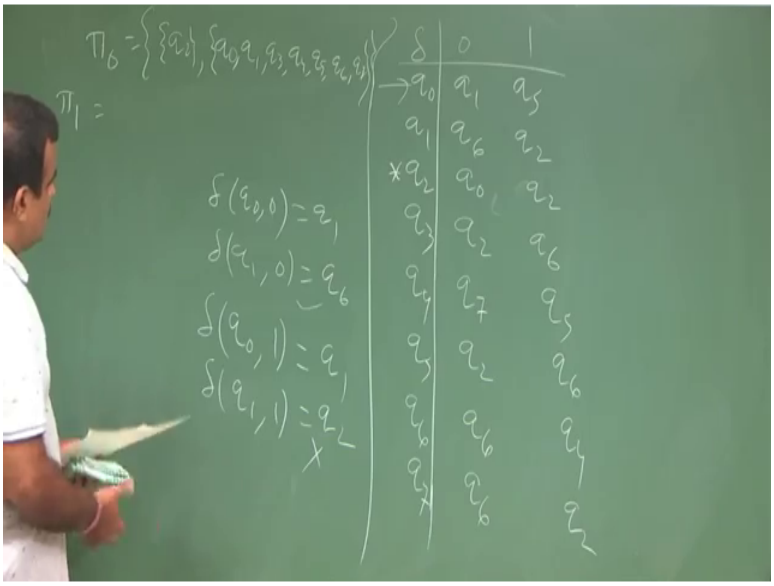
And we want to minimize this automata, this is having huge number of states. You can draw this, you are having huge number of states we want to just 8 state and we want to reduce this. So, how to reduce? So, first we will use the pi 0. So, pi 0 is the 0th level; 0th level, 0th equivalence class. So, 0th equivalence class is nothing but we have a final state which is q 2 and the remaining states are there. So, q 1 sorry q 0 q 1 q t is upon q 3, q 4, q 5, q 6, q 7. So, these are the our these are our first level partition.

Now, in the second level we have to form pi 1. So, to get pi 1 we just check what we just check with that. So, if say q 0 and q 1 you want to check whether they will be in the same class in the second level; that means, in the 1th equivalence class. So, that for that what we need to check? We need to check delta of q 0 a, this is where it is going and delta of q 1 a, where it is going; these two must be.

Student: (Refer Time: 23:08).

These two must be in the same class of pi 0 ok. So, this is the; this is the same class of pi 0. So, there has to be the there has to be the 0th equivalent.

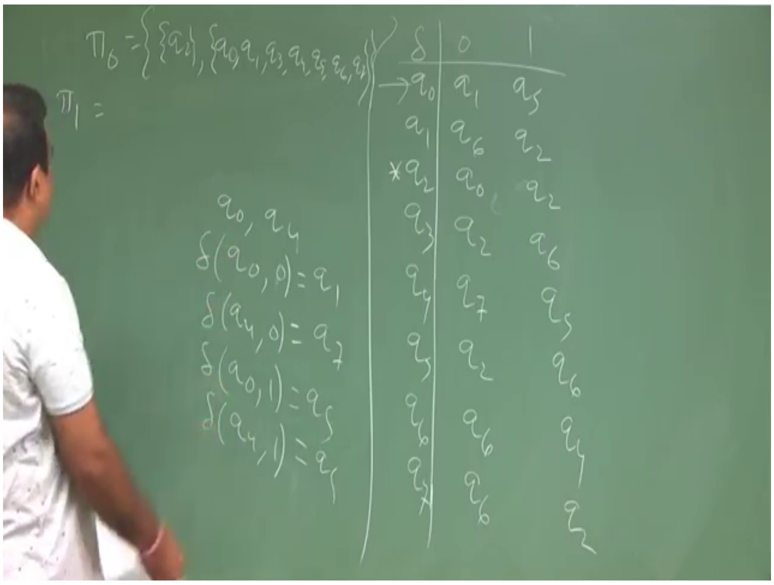
(Refer Slide Time: 23:29)



So, how to check that? So, you can check that delta of q 0 1, q 0 0. Delta of q 0 0 is what? q 1 and delta q 1 0, q 1 0 is q 6. So, q 1, q 6 both are same class, so, it is with this. Now, we have to check the for another symbol, delta of q 0 1, q 0 1 it is going to q 1, but delta of q 1 1 it is going to q 2.

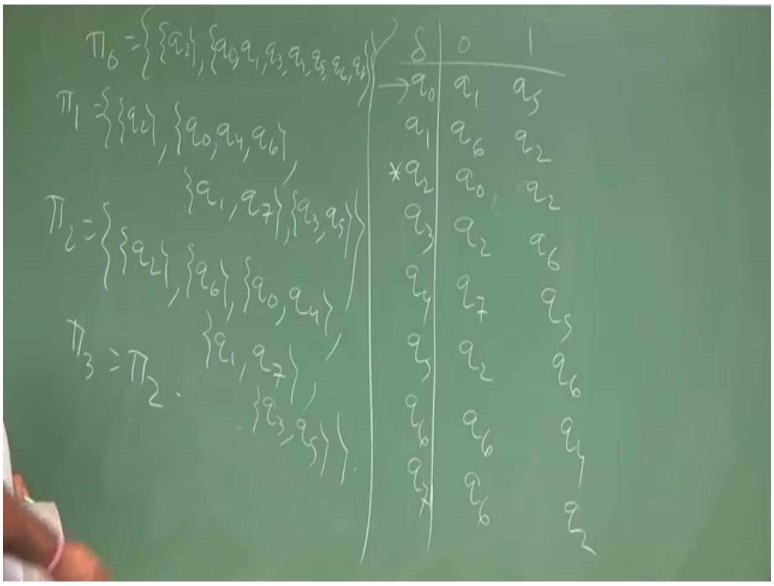
Now, q 1 is in this class, q 2 is in this class. So, they are not the 0th equivalence. So, they are not the, I mean if it is k plus 1th level they are not the kth equivalent, so that means, these two cannot be in the same class.

(Refer Slide Time: 24:25)



So, but if we can check q 0 and q 4, so q 0 and q 4. So, delta of q 0 1, I sorry 0 it is q 1 and delta of q 4 0 it is q 7. They are in the same class and delta of q 0 1 it is q 5 and delta of q 4 1 it is also q 5. So, they are in the same class; so, this is ok. So, they will come together.

(Refer Slide Time: 25:05)

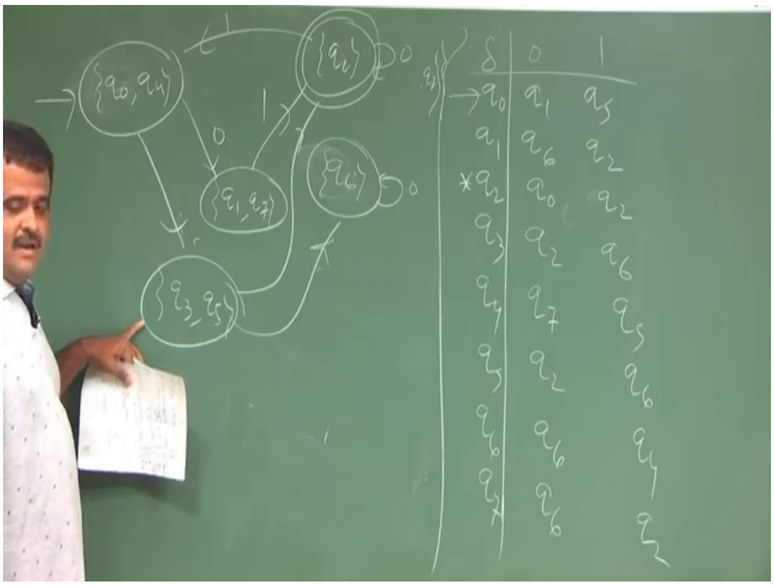


So, if you continue like this by checking each of these states, eventually we will get like this; in this level, so q 2 will be along and this will be q 6 sorry, this will be other next this level it will be q 0, q 4; q 0 q 4 you have check you have to check q 6 also, then q 1 q 7 these are in the same level, then q 3 q 5 ok. So, these are pi 1.

Now, we can continue with this for pi 2, we will check whether we can further divide this. So, q 2, so we can check q 6 and q 4. So, q 6 and q 4, so delta of say q 4 0 it is basically q 7 and delta of q 6 0 is q 6. So, q 7 and q 6 are not in the same not the one of the equivalent. So, they cannot be in the same class. So, if you check with this, we will be getting this q 6 is separated and q 0 q 4 will come together, this can be easily verified and q 1 q 7, q 1 q 7 and then we have q 3 q 5, so, this is the state.

Now, again we will further try to divide it if you try that we will get the same thing. So, this is converging; so, this is converging, then we stop here because we cannot further divide it. So, these are the final state of these. So, we can draw this, so, this is the final state of this, so, we can draw this.

(Refer Slide Time: 27:19)



So, these are the final state you want to draw this q 2 or q 2 is the final state we can take q 0, q 4, this is one state and this is q 2 this is another state and we have q 6; q 6 which is another state and we have q 0, q 4 and q 1, q 7 another state and we have q 3, q 5 another state. So, we have only 5 state, it was having 7 state now it is reduced to 5 state.

Now, we can have arc. So, q 0 this is the starting state, so this is going to with 0 input it is going here, with one input it is going here and from here we can come here with 1, with 0 input it is here this is the final state and from here with 1 we are going here and from here with 0 we are going here and with 1 we are going here, so, this is with 0, like this. Anyway, you can complete through. So, this is the reduced gamma, I reduce DFA coming from this, it has the less number of state. So, we will continue this in the next class.

Thank you.