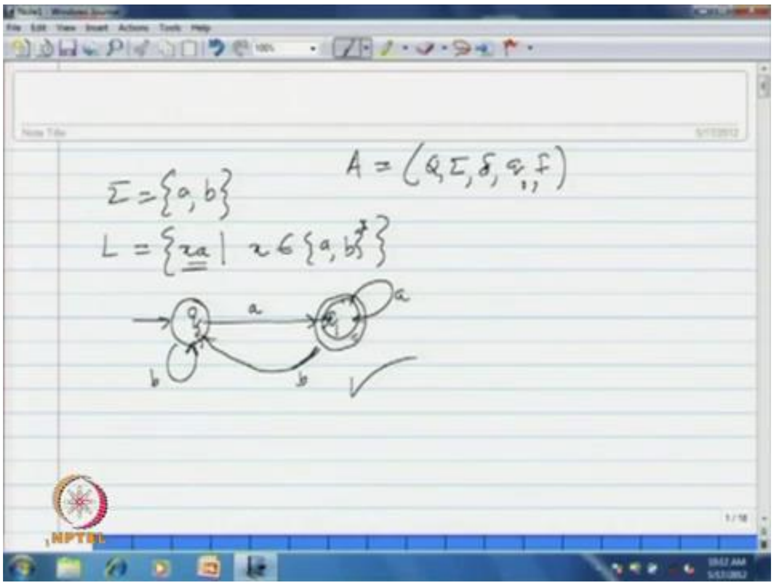
Formal Languages and Automata Theory
Prof. Diganta Goswami
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Module - 3
Finite Automata
Lecture - 2
Nondeterministic Finite Automata

We have already discussed and introduced deterministic finite automata, so today we will discuss another kind of automaton that is called Nondeterministic Finite Automaton. Before that we will just introduce one or two examples of deterministic finite automaton and then we will pass to describe formally nondeterministic finite automaton.

(Refer Slide Time: 00:47)



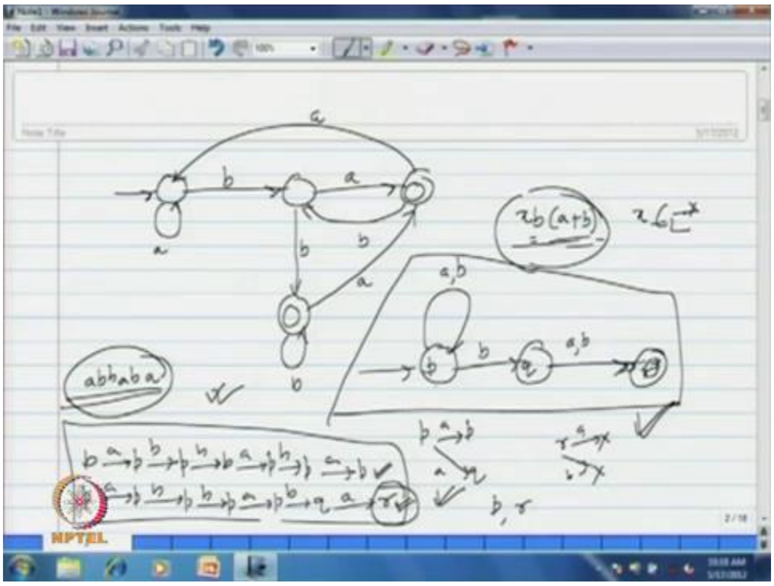
We have said that a DFA is quintuple, so A is a DFA the quintuple, Q, sigma, delta, q 0 and F, so where Q is the set of finite sets, sigma is a set of alphabet, delta is a transition map, q 0 is a star state or initial state and F is a set of finite set. Now, just consider the language over a alphabet, say containing two symbols say a, b just consider language over this alphabet, set of all strings over a, b which ends with a that means is it a form x a such that, x belongs to a, b whole square.

So, x is existing over a, b and eventually standard which a, so how to construct a DFA for this language. Now, if we observe that every string ends with a symbol a, so if you assume a state q 0 to be the initial state, then an occurrence of a has to be distinguished in the state q 0. So, this can done by making a transition found this state on input symbol a to some other set, so it is q 1, so on symbol a at the initial set q 0 will transit to state q 1 just a distinguish the occurrence of the symbol a.

Now, whereas, on symbol b we will continue to be in q 0 itself, because we did not distinguish the occurrence of b in the state q 0; now on subsequent a at q 1 thus rewind in the same set q 1, because it does not matter at it will still end with a. Now, by making q 1 as the final set all such strings can be accepted, now if you encounter a b at q 1, so it is a b at q 1, so we need to go out of q 1 as q 1 is a final set.

And we do not want to terminate the string with b and once you go out from q 1 on symbol b, again we need to see ((Refer Time: 03:43)) and a has accord again on all, but since q 0 is doing the same job. So, therefore, on b we can directly go to q 0, so that after that again we can check whereas an a has occurred, so therefore this will cover all possibilities and we will see that this D F A will accept the language that we have that given; that means, the set of all strings over a b that end with a.

(Refer Slide Time: 04:25)



Now, let us consider under D F A, where it has sets like and initial state, this state on b goes to the next state and on a it remains the same state, so this state on a goes to the another state again which is a final state, and this on b again goes to another final state.

Now, this final state on b goes to this state and this on a goes to the initial state and this state on b remains on the same state, whereas this on a goes to the other finite state.

So, unlike the example that you have just given is little tricky to find out or determine, what is the language accepted by this D F A. Now, but if you spend some time it may possible to conclude that a language accepted by this D F A is nothing but, the set of all strings over a, b which last, but one symbol as we depends it extends the strings of the form x, b and then a or b. But, x is any string over the sigma, depends over a, b, so any string concurrent should be and then concurrent to either a or b to be accept it by this D F A, but this not, so easy to comprehend the language x and y in this D F A.

Rather, if this language is even, then it is quite easy to construct a traditional diagram of an automaton which will accept this language, for example say we start with the initial state say b, this on a and b remain on the same state, on b it will go to state q, q on a and b will go to state say r, which is a finite state and there is no transition out of it. So, this type of automaton for a finite state there may be multiple transition, that means from state p we have transition on a to b itself and to q, so we have multiple extends on state b.

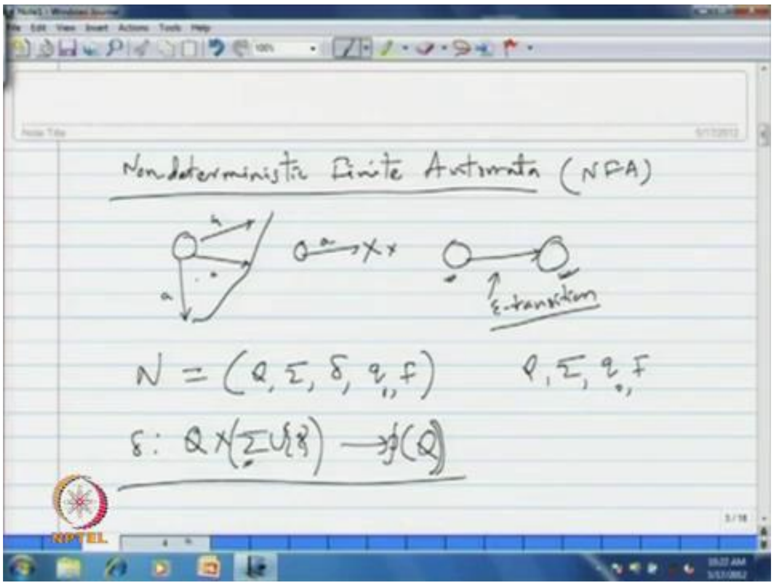
Then we do not have any transition out of state r on any input symbol, neither on a nor on b we do not have any transition out of b, so this slightly different concept to the automaton that we have already done, that is the D F A. So, in this automaton if say a b b, a b a is given as input, then we can trace we can find out two existence traces for this string. For example, on input symbol a and b we can always remain at b, we take a remain at b, we take b remain at p, we take b remain at p, we take a remain at p, we take b remain at p and we take a remain at p.

Again we can keep on doing the same thing on a remain at p, b remain at p, b remain at p, a remain at p and whenever you get this b from p to q we can have a transition on b and finally, from q on symbol a, we can go to the state r. So, clearly these are two possible axiom traces on the input string a b b, a b a for this finite automaton, therefore we have two next states p and r on the input string a b b, a b a for this finite automaton.

Now, since for this input string, we have two eligible traces and we needs in one case the state p, the next state is p and for other case the next state is r and here r is a finite state, if you assume that in such a case in one of the axiom traces, when you are after completion the state edits is a finite state we say that just accepted. If you concentrated this notes on acceptance, when you say that this string will accepted by this finite automaton.

So, this type of automaton which have external features, where we have multiple n extents or where there never be any next is on sin two symbol and with some more external features is said to be nondeterministic finite automata. To what will do, will now formally introduce the notes on nondeterministic finite automata.

(Refer Slide Time: 11:03)



In contrast to D F A while we have a unique, so I going to be introduce nondeterministic finite automata or simply N F A, so in contrast to D F A where we have a unique next state for a condition, from a state on an input symbol. Now, we consider a the finite automaton with nondeterministic transitions that means, the transition is nondeterministic, if there are several transitions from a state out away single symbol say a.

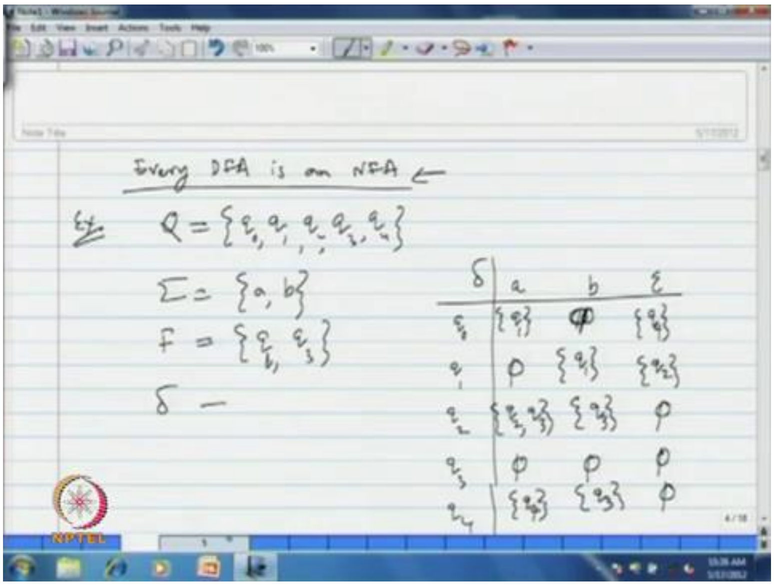
They may be multiple transitions are same symbol a or possibly there will be 0 transitions that means, there may not be any transition out of a state on some symbol a, there may not be any transition. So, either multiple next states or no next state and then there may be a transition out of a state without taking any input to some other state, this type of transition is said to be an epsilon transition.

So, those are possible nondeterministic transition there are going to be introduced in non deterministic finite automata, so formally in the line of deterministic finite automaton, we define a nondeterministic finite automaton or N F A as a quintuple. So, N is a quintuple again is Q, sigma, delta, q 0, F, where Q, sigma, q 0 and F that having all same meaning as you defined case of the D F A and we simply differentiate this delta that condition met for this N F A.

So, what we have said is that in case of N F A, we can go from one state on some input symbol say sigma to multiple next states, so this is not a single state now, in case of D F A it was simply Q, the transition may be defined like this Q cross sigma goes to Q. So, in this case it may go to multiple next states or they are not may not be any next state that means, it is a any sub set of Q that means, it is a power set of Q which introduced as or denoted as p Q is a power set of Q.

And since we take not only a symbol from sigma, but it may be an epsilon also, because there may be from epsilon transition without taking any input, there may be transition out of any state from one state to another state. So, therefore, is Q cross sigma union epsilon, so the transition map is defined as Q cross sigma union epsilon goes to the power set of Q, so that is how we defined it transition.

(Refer Slide Time: 14:49)



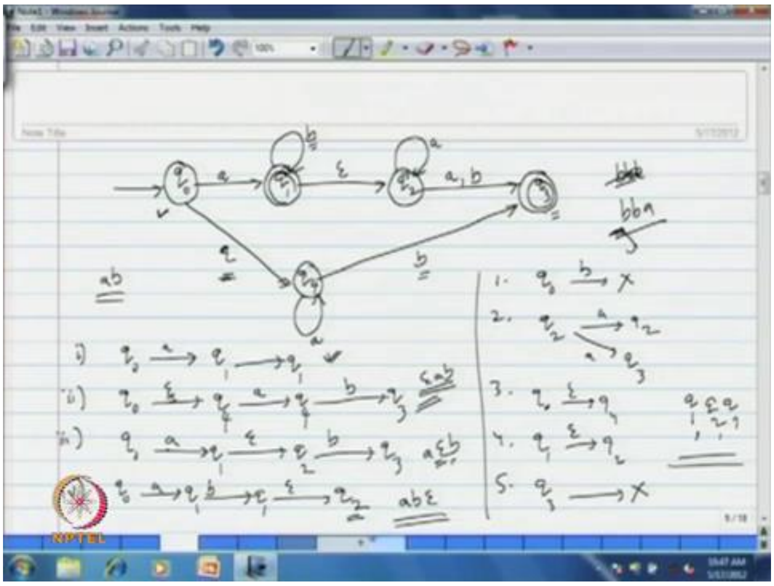
Now, one thing is very clear is that, every D F A can be treated as N F A, because D F A is a special case of N F A, so this is quite clear from our definition. Now, let us give an example, consider a N F A where the set of states is specifically q 0, q 1, q 2, q 3 and q 4, there are five states sigma is a, b conducts two symbols, the set of finite state is specifically q 1 and q 3.

And delta is defined by, therefore lined up just define delta by during this map, so in this case we have five steps q 0, q 1 q 2, q 3 and q 4, so for input symbol a and b we will have two columns, besides that we will have need to have an column for epsilon to consider the for epsilon transition. Let us say q 0 on a transit to q 1, so it is a subset containing only state q 1, so q 0 on b there is suppose no next ((Refer Time: 16:35)), so it is an empty set phi say q 0 on epsilon, suppose it goes to state q 4.

So, it is as subset containing, the state q 4, say q 1 on a does not have any transition that is fine, so q 1 and b say it remains the q 1 containing the subset containing only state q 1 and q 1 on epsilon, suppose it goes to q 2, the subset containing the state q 2. Similarly, suppose q 2 on a it goes to two states, there is two transitions defines a multiple next states q 2 and q 3. So, it is subset containing two states q 2 and q 3 say q 2 on b, suppose it goes to q 3 and q 2 on epsilon, suppose it goes to no other states.

So, it is phi similarly on q 3, there is no transition defined on any input symbol neither on epsilon, so it is all phi, similarly q 4 on a suppose it remains in a same state, q 4 on b, suppose it goes to q 3 and q 4 on epsilon no transitions define. Suppose these are transition map that you have, so the way we have drawn the transition diagram already for D F A.

(Refer Slide Time: 18:33)



In this case also we can draw a transition diagram for the N F A, q 0 is the initial state, so q 0 on a goes to q 1, b no transition defined and on epsilon it goes to q 4, so there are two transitions on a and epsilon. Suppose, on a it goes to q 1 that have given and on epsilon it goes to q 4 and q 1 is also a finite state, again as given in the transition map q 1 on b remains in the same state. And q 1 on epsilon goes to state q 2 and there is no transition defined on a for state q 1, so q 2 on a remains in the same state as given in the transition map.

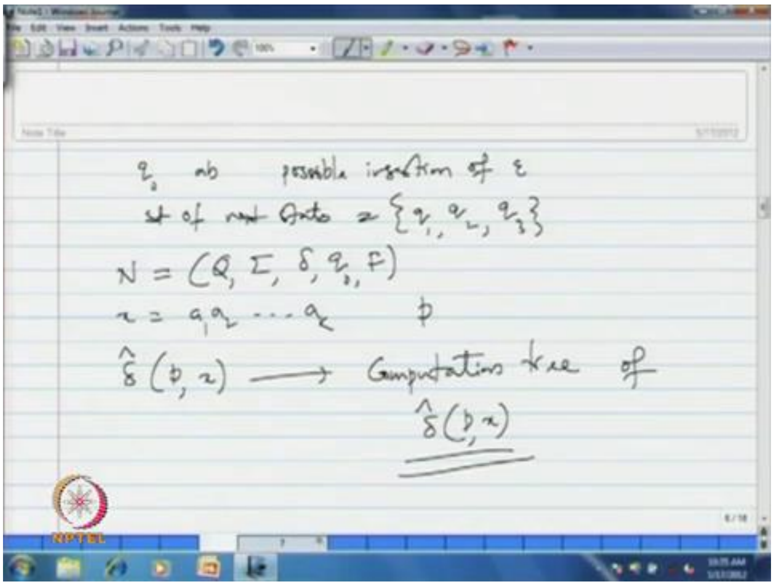
And q 2 on a again goes to the state q 3 as defined multiple next states for q 2 on input symbol a, q 2 on b it also goes to q 3 and on q 3 we do not have any transition defined or any input symbol. And then q 4 on a remains in the same state q 4 and on b it goes to the other finite state q 3, so these are transition map that we have drawn for example N F A, now just consider, what are the non transition that we have in this particular N F A.

So, nondeterministic transitions are that there is no transition from q 0 on input symbol b there is no transition defined, then their multiple transitions from q 2 on input a, the two transition defined on q 2 and q 3. And then their epsilon transitions from q 0 to q 4 on epsilon and q 1 to q 2 on epsilon, q 0 to q 4 on epsilon and q 1 to q 2 on epsilon, so these are the various possible nondeterministic transitions that we have, similarly on at q 3 there is no transition defined on any input symbol.

So, there is all possible now transitions that we have in this example, now just consider the string a, b and there is process the string a, b at state q 0 and let us see what is the execution impress, in this case. We see that in process the string a, b at state q 0, from q 0 taking a it may go to state q 1 and at q 1 taking b it may remain in the same state q 1, so this is one possible as you can trace. Then the next one state at q 0 without taking any input, so this is the epsilon transition that we have, it can go to state q 4, at state q 4 taking input symbol a, it remain in the same state q 4.

And from q 4 taking this transition, it may go on input symbol b to state q 3, then the third possibility it is that q 0 taking a it can go to q 1 and then at q 1 taking an epsilon transition you can go to q 2 and at q 2 taking b on input b, it will go to state q 3. So, possible axiom trace for a given string a, b at state q 0, 100 percent trace is that, that q 0 will take the input a go to state q 1, then it will take input symbol b it will remain the same state q 1. Because, on q 1 on input symbol b it may remain in the same state and then can take epsilon transition at q 1 and it will go to state q 2 and see that, the three distinct states q 1, q 3 which are feasible from q 2 by other string a, b.

(Refer Slide Time: 24:52)

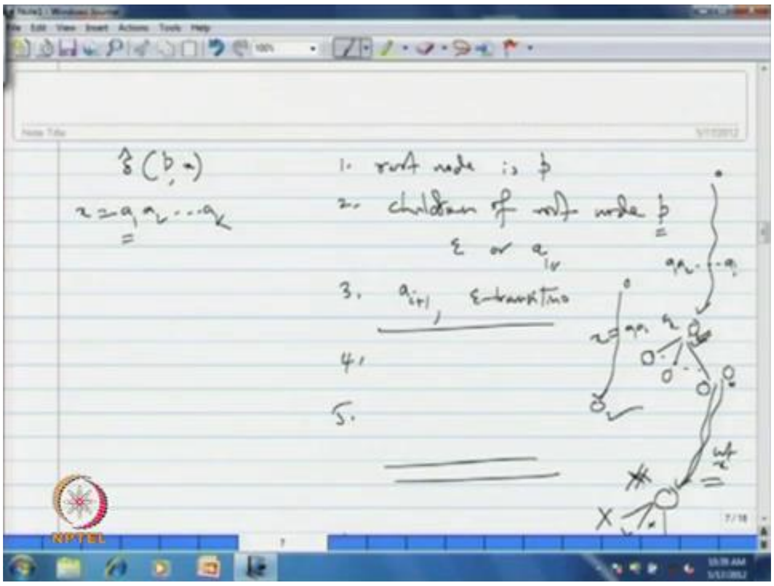


That means, while tracing a path from q 0 on string a, b what a consider is that, we consider possible insertions of epsilon industry that means, wherever is epsilon transition defined. That means, in the second case what we have seen is that, you have considered the string to be epsilon a, b, because you have taken first an epsilon move and then you have taken the move on a on q 4 and a move on b at q 4.

In the second case you considered string to be a epsilon b, because first you have considered a move on a at q 0, then you have considered epsilon move, because you defined a at state q 0 at q 1 and you have moved to q 2, and at q 2 you have gone to b, q 2 have gone to q 3 on input symbol b. Similarly, in the last case you have considered string to be a, b, epsilon, because in the last state in epsilon transition, because this is defined and you have taken accordingly and moved to the state q 2.

So, wherever the epsilon transitions are defined we can take it and hence, we can consider possible in transitional epsilon at various positions on that string. So, it is clear that if you process the interesting heavy the state at q 0, then the set of next is sets of next are q 1, q 2 and q 3 now just consider N F A, Q, sigma, delta, q 0 and F. Now, given an input string x out of form say a 1, a 2 up to some symbols a, a k and an input state and the state of the N F A state is p, the state of next is starting with p we process the string x, then we get a set of next we defined as delta hat p x. Now, this can easily be computed by using a tree structure, which is called computation tree, computation tree of delta hat p x, so let us for define quality computation tree for delta hat p x.

(Refer Slide Time: 28:15)



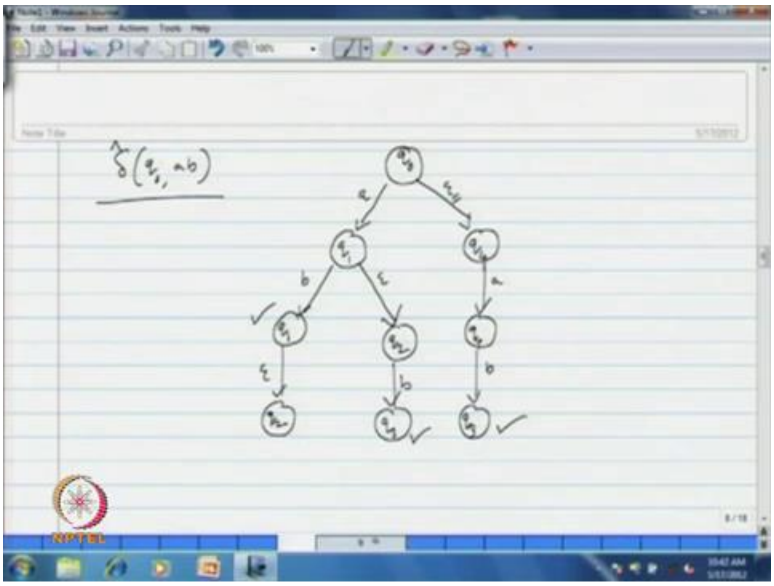
So, if we consider delta hat p x, while x is a 1, a 2 up to a n, you can then the computation for this delta hat p x is basically a tree, which is rooted at node p, so root node is p, then children of root node p a precisely those nodes which are reachable or which are having transition from p, y i epsilon or the input symbol a 1, the first input symbol. We started p and then we see for any possible transition on epsilon or on the first input symbol a 1, so those nodes we need a children of p.

Then for any node, the branch from the root to did that node, if the branch is leveled which a 1, a 2 up to say a i, then the children of those nodes will be exactly those nodes, while we have a transition from this node to that node via the input symbol a i plus 1, because the next input symbol is a i plus 1, we have to go up to a k or via an epsilon transition. So, you need to consider already an epsilon move from this node or any move on input symbol a i plus 1, so all those nodes we need a children of this particular node.

Then if judge a finite state, whose branch from the root, suppose this branch from the root to this node is level x, which is equal to a 1, a 2 up to a k, then we simply put a tick mark. And finally, if the level by branch on the leaf node, suppose you are concentrate tree and eventually write a leaf node, because there is no any transition defined on any two symbol from that particular node.

So, therefore, ((Refer Time: 31:05)) the leaf node and the level of this branch is a prefix of x is not x, but it is a prefix of x, then you put a cross mark to indicate that. We cannot could not process the full string x starting with x following this part, so that is how have defined a computation tree.

(Refer Slide Time: 31:40)

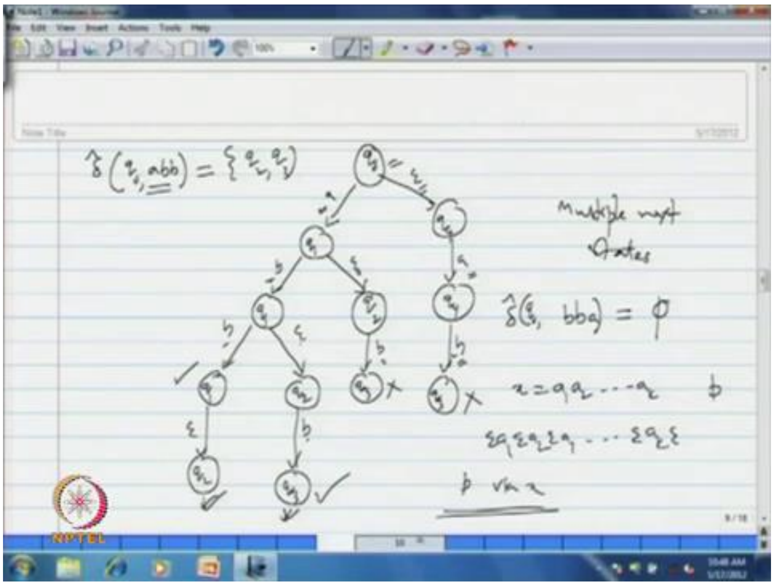


Now, let us draw the computation tree for the string a, b that means, for the string a, b starting at q 0 for a given example N F A, the example N F A was this one ((Refer Time: 31:53)). So, starting at q 0, we can process the string a, b in a different way that we have already given over here, so accordingly we can start at q 0 is a root node, and you can process the string a, b following this part take a we arrive at node q 1 or you can also follow an epsilon transition to arrive at node q 4 starting at a.

If we consider a input b we will move to state q 1 again, because on b q 1 remains the same state and since, q 1 at this q 1 we have processed the full string, we put a tick mark over here and then on epsilon it goes to state q 2. Similarly, at state q 1 taking an epsilon transition you can go to state q 2 without taking any input, so far we have processed only a in this branch and at q 2 taking b you can arrive at the state q 3, so here again you have processed the full string a, b around this path, so you put a tick mark over here.

And where is q 0 on epsilon it goes to q 4, at q 4 taking the input a it remains in the same state q 4 and at q 4 taking b you can go to q 3, and there is no transition defined on q 3 and you have already processed the full string a, b over here, so you put a tick mark. So, that is how we have drawn the computation tree, how does string for delta hat q 0 a, b.

(Refer Slide Time: 34:21)



Similarly, you can also draw the computation tree for the string a b b say delta hat q 0, a b b, so you follow the same process, then what you see is that, the computation tree looks like this one q 0, on a it goes to q 1, taking b it goes to q 1 again and taking b it goes to q

1, remains the same state. And at q 1 you can take an epsilon transition which will go to q 2, but we have already processed the full string over here a b b, therefore you put a tick mark over here.

Similarly, taking first a it will go to q 1 and then taking an epsilon at q 1 it can go to q 2 and q 2 taking a b, it can go to q 3, but q 3 no further define, so therefore if you follow this path a, epsilon b a process only a, b not the full string a b b, so therefore you put a cross mark over here. Similarly, if you follow this path a, b, then at q 1 again you can take an epsilon transition to go to state q 2 and q 2 taking b you can go to state q 3.

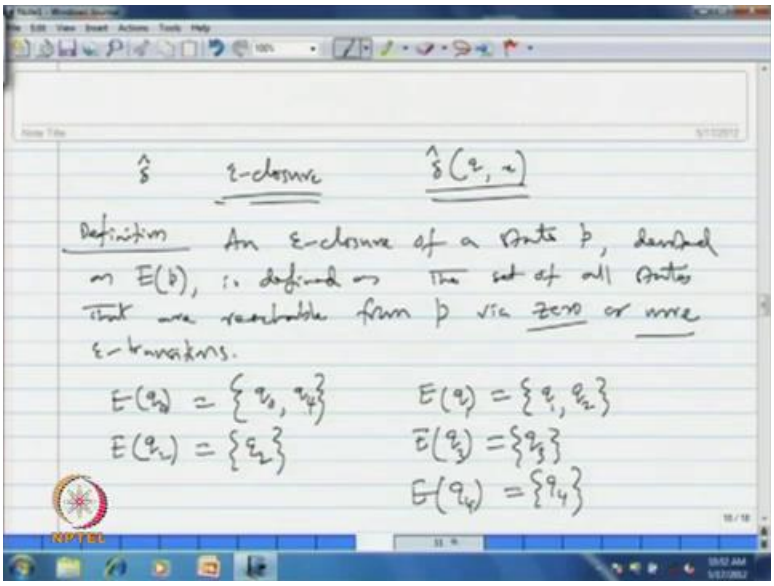
Now, you see that no further transition defined on q 3 and the string that you have processed starting from q 0 is q 0 to this q 3, and it is path is a, b, epsilon b that means, a b b, so we have processed the full string, so therefore you put a tick mark over here. And then at q 0 initially you could have taken an epsilon transition and you could have gone to state q 4 at q 4 taking a, you can remain the same state q 4, and taking input b you can go to state q 3, while no further transition is defined.

So, therefore, the string that you have processed starting from q 0 is epsilon, a, b it is not the full string a b b, so we put a cross mark over here. Now, as the automaton given in the example is nondeterministic, if a string is processed in a state, then they may be multiple next states, that we have already seen. For example, if you process the string a b b at q 0 the multiple next step of possible next step are q 2, q 3, q 2 and q 3 only, so delta hat q 0, a b b is q 2 and q 3.

Similarly, if you process the string say b b a just that go to the N F A, they process the string b b a, then there is no any transition defined from q 0 on input symbol b, the only possible way to transit via this epsilon transition to q 4, on q 4 and the input small b we can go to q 3. So, you can process the first b and then will go to state q 3 once you arrive at state q 3 there is no further transition define on any input symbol, hence after processing the first b we cannot process further.

So, therefore, for this case delta hat q 0, b b a is empty, because you could not process the string, thus given a string, so x equal to a 1, a 2 up to a k and the state say p of the automaton. Now, by treating x as epsilon a 1, epsilon a 2, epsilon a 3, so on epsilon a k epsilon that means, in certain epsilon in between every two symbols and at the end and at the beginning, by looking at the possible complete branches starting at p, we find the next states for p via x.

(Refer Slide Time: 39:38)



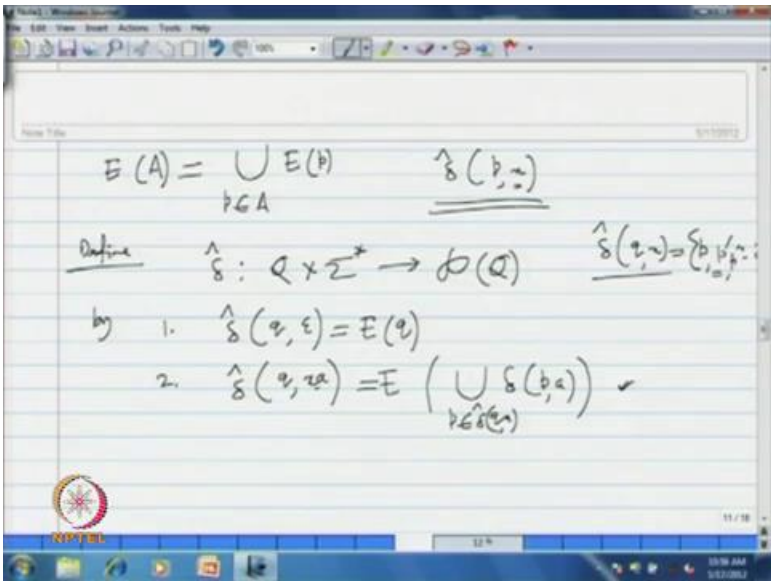
Now, to introduce the notion of delta, delta hat that means, the extended transition function in a N F A, we first introduce the notion of epsilon closure. So, we need to define what is the extended transition function for an N F A, to define this formally we need to first define for it is epsilon closure, so let us first define epsilon closure. So, an epsilon closure of a state p is noted as E p, epsilon closure of p is defined as the set of all states that are reachable from p via 0 or more epsilon transitions.

So, just follow zero or more epsilon transition, whatever states are reachable from p, let us have a state say to be epsilon closure of the state p for example, in our example N F A that we are given here ((Refer Time: 41:27)), the epsilon closure of q 0 will be following it is epsilon transition from q 0, and arrive at q 4. And without taking an input may also remain in the same state q 0, therefore the epsilon closure of q 0, since without taking any input remain in the same state q 0 and there is a epsilon transition defined on q 0 to q 4.

So, these are only two personal states, it can reads from q 0 following 0 or more epsilon transition, q 0 is following zero epsilon transition, q 4 is arrived at where following one epsilon transition, so this is the set of possible states that you arrive at, so therefore epsilon closure of q 0. Similarly, you can define epsilon closure of q 1, you can arrive at the state q 1 without taking any input that means, zero epsilon transition and then by taking one epsilon transition you define on q 1 it goes to q 2. So, this is the set of possible states following zero and more epsilon transition on sate q 1 and hence is the epsilon closure of q 1.

Similarly, if you look carefully you find it your q 2 is the single state q 2, because there is no epsilon transition defined on it on taking 0 epsilon move will remain the same state q 2, therefore it only possible state that you have. Similarly, E q 3 epsilon transition q 3 is does it containing q 3 only and epsilon closure of q 4 is a set containing q 4 only, so according to definition we have computed, the actual closure of which and every state for example, N F A that they already given.

(Refer Slide Time: 43:51)



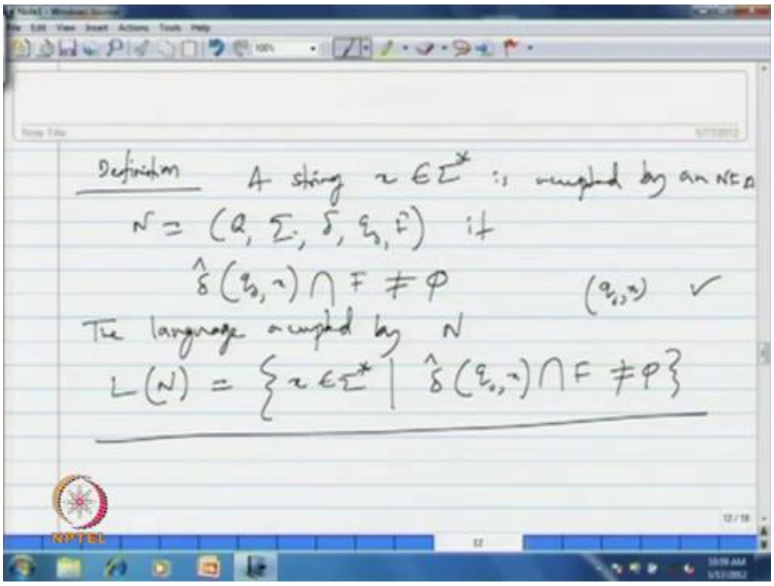
Now, is a state is given say, A is a state of this, we can also compute the epsilon closure of this state, by first computing epsilon closure on each and every state in A and then taking union. That means, epsilon closure of the state, set of states of A is nothing but, union of all epsilon closures of states p, where p is a state that belongs to a, so that is how we define epsilon closure of a set of states a, once you have defined this epsilon closure, when you are ready to formal define the set of next state by string.

That means, delta hat p, x or the external transition function for an N F A, so define delta hat p x or external transition function for N F A and this, so is a map from q cross this x any string from sigma star, sigma star, it goes to power set of q, it may be empty for any subset of q, by giving two steps. The first step delta hat q epsilon without taking any input or taking zero or more epsilon moves, whatever states this step is given by epsilon closure of q and then delta hat q x a, so x a, x is any string over sigma star and delta hat q x a is defined as...

So, what you do is that you see what are the possible states delta p a, where p is a state that you can arrive at by processing the string x stating at q that means, delta hat first you have compute delta hat q x. So, then we state of steps is a suppose p next say this p dash and so on, now you see the next symbol is a, so one step we can arrive at, but processing a at p, where p is a state that belongs to delta hat q x.

And you take the union, then for all possible p that you have in delta hat p x, for p p dash, so p double dash you compute this little for p belonging to delta hat q x for every p. So, once we find out the state that you can arrive at on symbol a for every p belonged to delta hat q x, we take the epsilon closure of this, because there may be again further epsilon condition possible from that state.

(Refer Slide Time: 47:49)



So, after defining extended transition function, now you can define the acceptance of a string and the language over N F A, we say that a string x belonging to sigma star is accepted by an N F A. So, N which is Q, sigma, delta, q 0, F, if delta hat q 0 x starting at q 0, if you process the string x, so we will get a set of states and in that set of states, if you have a finite set, then we say that string is accepted. That means, if delta hat q 0 x and f they have a common element that means, if this intersection is non empty we say that the string is etcetera.

So, in context of computation tree of q 0 x just be a final state among the nodes marked, ticked, so everywhere you have marked tick, if there is a node which is a final state, then we say that the string is accepted. Thus the language accepted by the N F A N is defined as the language of N, the collection of all those strings x belonging to sigma star such that, delta hat q 0 x and F has common element at least, so that is how we have defined the language accepted by an N F A.