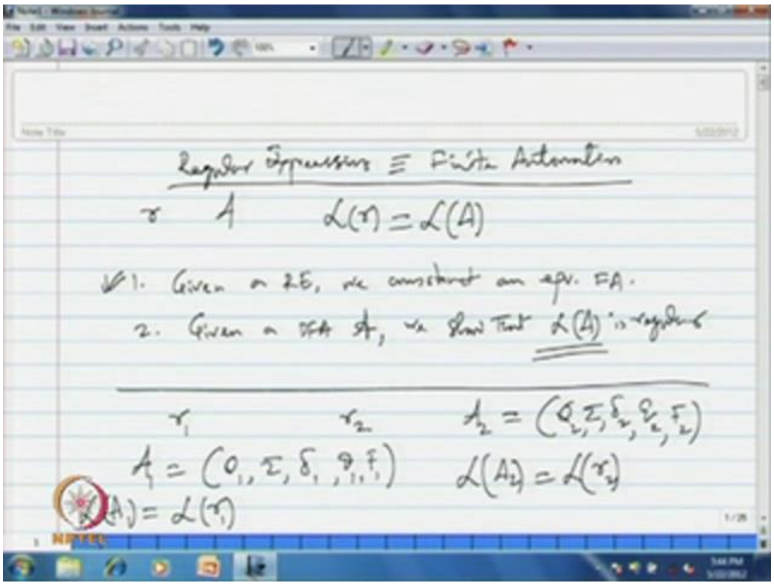
Formal Languages and Automata Theory
Prof. Diganta Goswami
Department of Computer and Engineering
Indian Institute of Technology, Guwahati

Module - 5
RL - RG - FA
Lecture - 1
RE – FA

So, you know that language represented by a regular expression is defined as regular language. Now, when a position to provide all kind of definition of regular languages via finite automaton; either d F a or n F a. And also via regular grammars; that is the class of regular languages precisely the classes of languages accepted by finite automata. And also it is class of languages generated by regular grammars. So, this results will give the providing with few theorems.

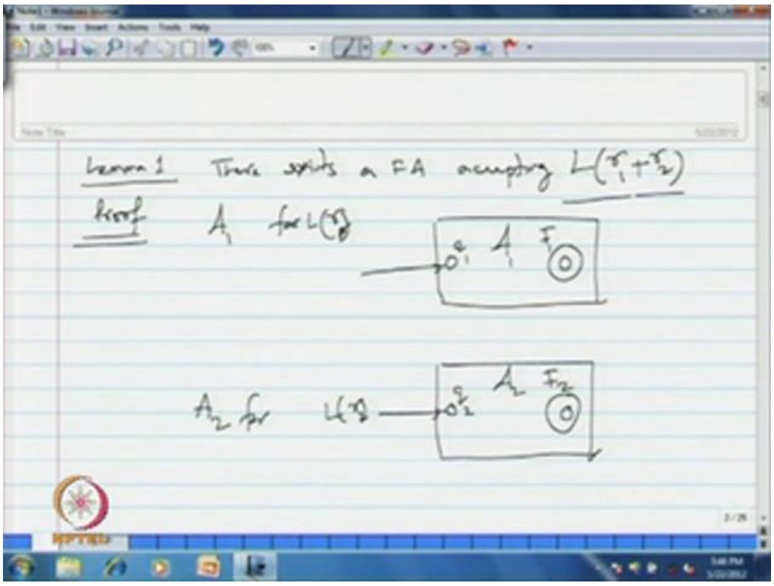
(Refer Slide Time: 01:11)



So, first we are going to proof that regular expressions are equivalent to finite automaton. That means the class of languages accepted by D F A or N F A is same as represented by regular expressions. So, we say that a regular expressions r is equivalent to your finite automaton A. Suppose A is finite automaton, which says that r is equivalent to the finite automaton A; if the language represented by the regular expression is precisely expected by the finite automaton A. Now, in order to proof this equivalence regular expression equivalent to finite automaton so what I will do? We proof these two. Given a regular expression R E, we construct and equivalent finite automaton. And then, given an D F A, A we show that L of A is regular, That means there is a regular expression R, that represents the same language accept by finite automaton A that is L(A).

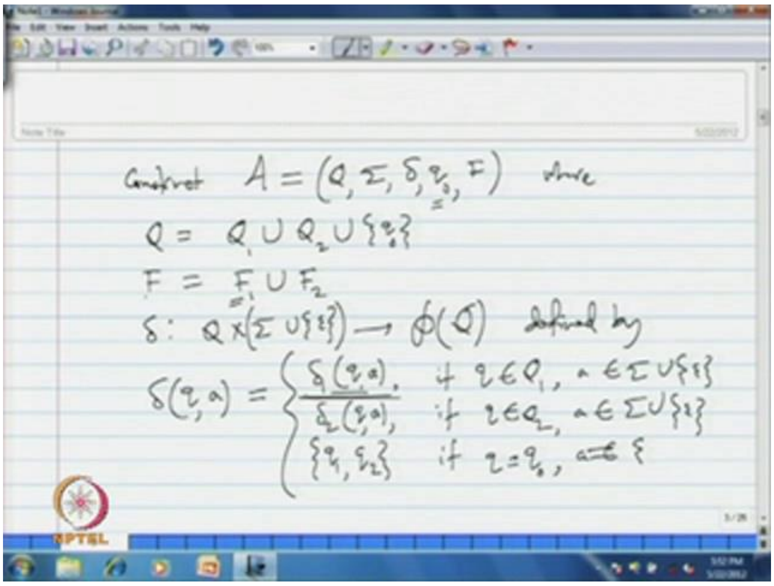
Now to proof 1 we will first proof this first point. So, we will first proof 3 lemmas so, what we will assume is that; so, r 1 is a regular expression, and r 2 is a regular expression. Then, let us assume that there exists finite automata say A 1 denoted as Q 1, sigma, delta 1, Q 1, F 1 and which accept the language represented by the regular expression r 1. That means; L of A 1 is exactly L of r 1, I assume that for this regular expression r 2. We have an automaton say A 2 given by Q 2, sigma, delta 2, Q 2 and F 2, such that; L of A 2 is exactly L of r 2. So, we assume that for 2 regular expressions given regular expressions; we have 2 automaton A 1 and A 2 respectively. That accepts the corresponding language of the regular expressions.

(Refer Slide Time: 04:52)



Now, let proof these lemurs so, the first lemmas is that there exists a finite automaton accepting L of r 1 plus r 2. That means; given r 1 and r 2 and the corresponding finite automata. Then, I can construct a finite automaton accepting a language r 1 plus r 2. Let us see how can I do that. So, let us assume that; the automaton A 1 for r 1, or L of r 1is this 1. So, here is star state which is Q 1 and there are many other states and eventually set of final states there is denoted as F 1. So, this is the automaton A 1, similarly; we have the finite automaton A 2 for L of r 2. Where, we have the star state Q 2 as we have already find, and set of final states this is F 2. So, this automaton accepts L of r 2 so, A 2 is for L of r 2. Now, from these 2 automaton will construct a finite automaton which is say a, which will accept the language L of r 1 plus r 2.

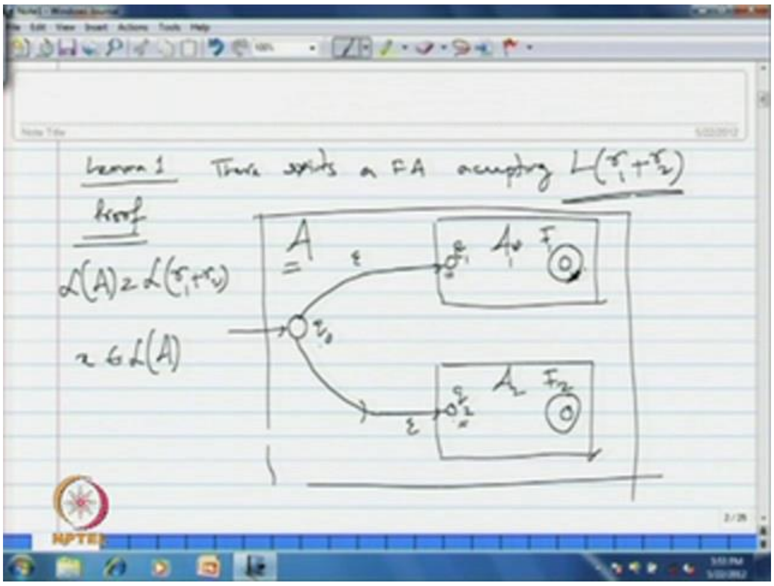
(Refer Slide Time: 07:02)



Now, to do that from A 1 and A 2 we construct say A, which will accept L of r 1 plus r 2. We claimed that way; so, it is nothing but Q, sigma, delta, q 0, F the corresponding elements. Where, Q is basically all the states in A 1 union states in A 2 and then, we introduce a new state we just star state would a automaton A there Q 0 is the new state. That we have been introduced then set of final states for this automaton A is the union of the final states of A 1 and A 2. And we defined delta; which is from Q cross sigma union epsilon this is basically n, n F A goes to the power set of Q.

So, we defined by, we defined this transition map by this. So, delta q a for this automaton where, Q is n state then, it maybe we keep all the traditions from the automaton A 1 that means delta 1 Q A. If q belongs to Q 1, and a belong to belongs to sigma union epsilon. That means; it retains all the transition functions of automaton A. It also retains all the traditions functions of the automaton Q 2. That means delta Q a equal to delta 2 Q A, if Q belongs to Q 2 because we have used delta 2 and a belongs to sigma union epsilon. Finally, from the star state Q 0; if q equal to q 0, if this a star state then, on epsilon the automaton A will transit to either Q 1, there is star state at A 1 or Q 2, that means; it is nothing but q 1 union q 2. So, if q equal to q 0, and A is equal to epsilon so, epsilon transition from star state of new automaton A; it will move to either the star state of A 1 or the star state of Q 2.

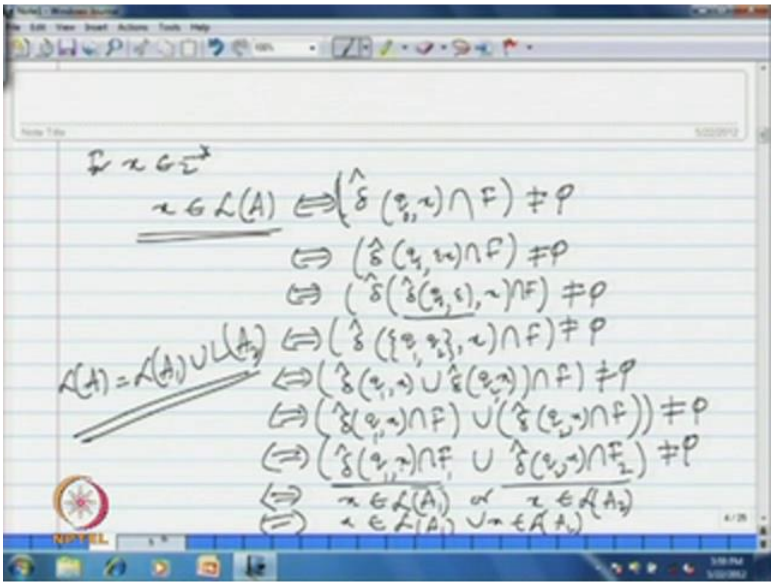
(Refer Slide Time: 10:30)



That means; in this figure what we do is we had automaton A, we introduce a new star state is q 0, and from this q 0 to the star state of A 1, we give epsilon transition. And, from the star state of A, we give epsilon transition to the star state of A 2. And, the result in automaton, that we have got the result in automaton. That we have got is the automaton A, and I claimed that this automaton A accepts the language represented by the regular expression r 1 plus r 2. That means; L of A is nothing but L of r 1 plus r 2.

So, intuitively it is quite clear, because if this automaton a accepts the language, accept the thing suppose x belongs to L of A then, it has to started into the process the string x its star gives 0. It must first I will transit to star state of A 1 or it may transit to the star state of Q 2. By taking an epsilon transition first without loss of generality, if it transits to a star state of A 1 that is Q 1, and from this point onward it will follow all the traditions of A 1, because you have retained all the traditions of A 1. And then, onward by process the string x eventually it will reach one of final states from F 1. And, since F 1 and since; F 1 is also a final state of this automaton F. The string accepted by the automaton A 1 as since x is process at star state Q 1, and eventually it enters a final state which is in F 1.

(Refer Slide Time: 13:20)



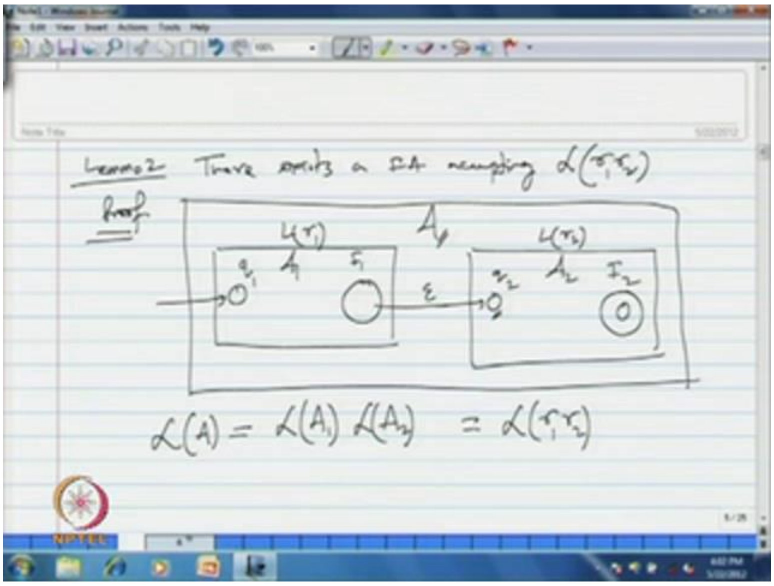
That means, formally it can write it that; for x belong to sigma star, x belongs to the language of the automaton A if and only if there is a transition, or it processes the string starting at q 0, delta hat q 0 x. And, eventually it arrives at final state that means; the set of next states intersection F not equal to phi. That is how we defined the acceptance why n F A. If and only if delta hat we can write it as q 0 epsilon x, because x can be written as epsilon x intersection F not equal to phi. If and only if delta hat since this q 0 epsilon x, we can write it as delta hat q 0 epsilon first process that string epsilon. And then it take the string x so, you apply the same x intersection function delta hat, this intersection F not equal to phi.

Now, the way we have defined the transition delta hat q 0 epsilon, this is nothing but it may go to either q 1 or q 2. That means; we will have delta hat 2 states q 1 and q 2, because we star state of the automaton A take epsilon intersection, it may either go to q 1 or may go to q 2. Then, it processes the string in x so, this intersection F not equal to phi. So, this means that; if and only if delta hat q 1 x and delta hat q 2 x and take the union delta we defined delta hat q 2 x q 2, x so, this intersection F not equal to phi. So, if and only if by applying laws of statuary; we can write it as delta hat q 1 x union sorry, intersection F union delta hat, from this delta hat q 2 x intersection F q 2 x intersection F this not equal to phi.

This if and only if since; this F over here delta hat we had considered only the moves of automaton A 1 delta hat q 1 x we started at star state of q 1 and from that onward there will not be any traditions from automaton A 2. So, therefore; we can write this as F 1 union and this one we can write it as delta q 2 x since from this onward we will take only the traditions from automaton A 2. Therefore, we can write it as F 2 so; this is not equal to phi. So, this if and only if this says that x belongs to language the automaton a, because star the star state of automaton A 1 process the string x. So, if you arrive at least sums that which belongs to the automaton, I mean final set of A 1. Similarly, this says that x belongs to L of A 2.

So, therefore; x belongs to either this or either L of A 1 or x belongs to L of A 2. So, since this not equal to phi delta hat q 1 x intersection not equal to phi means x belongs to L of A 1. Similarly, delta hat q 2 x intersections F 2 not equal to phi this belongs that means x belongs to L of A 2. This means; x belongs to L of A 1 union x belongs to L of A 2, therefore; if x belongs to L of A then, x must belong to either L of A 1 or x belongs to either L of A 2. Therefore, L of A equals to L of A 1 union L of A 2, so this put.

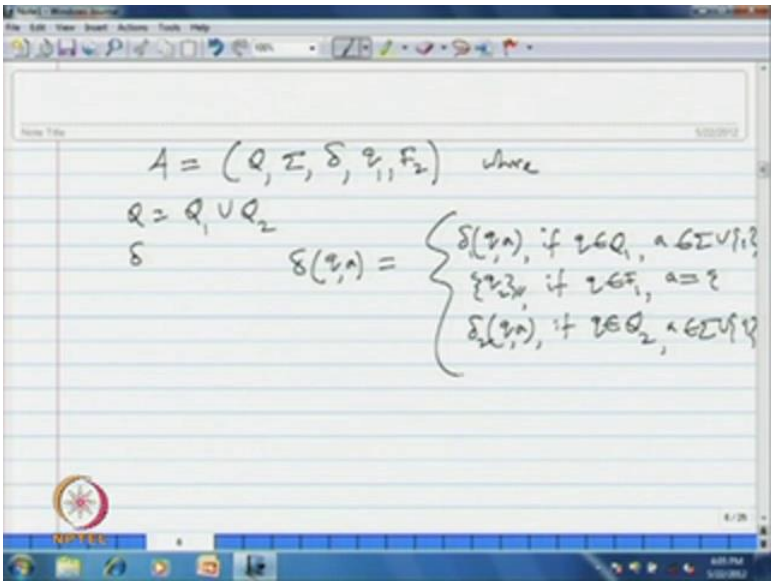
(Refer Slide Time: 20:07)



Now, in lemma 2; we will show that so, if there exists finite automaton for regular expression r 1 and r 2. Then, there exists a finite automaton accepting L of r 1 r 2, where the concatenation of the 2 regular expressions, let us proof it. So, it looks quite simple and similar to the previous one. So, what we do if this is the automaton A 1 with star state q 1, and the set of final states F 1. And, this is the automaton A 2, this is for L of r 2 and this is for L of r 1, which has a star state q 2, and the set of final states F 2. What we do in the automaton A? That you construct for L of r 1 r 2 from this 2 automata A 1 and A 2. We consider this F 1 to be F 1 the set of final states to be non final states, and give epsilon transition from each of these final states to the star state of automaton A 2.

And, in a delta new automaton A that you have constructed q 1 in the star state, and F 2 the set of final states of A 2 will be the will also be a final state of A. And all these final states of q 1 will be none final states or ordinary states in A. Now, we claimed that; the language accepted by the automaton A is nothing but the language accept by A 1 concatenation the language accept by A 2. Therefore, this is nothing but L of r 1 concatenation r 2.

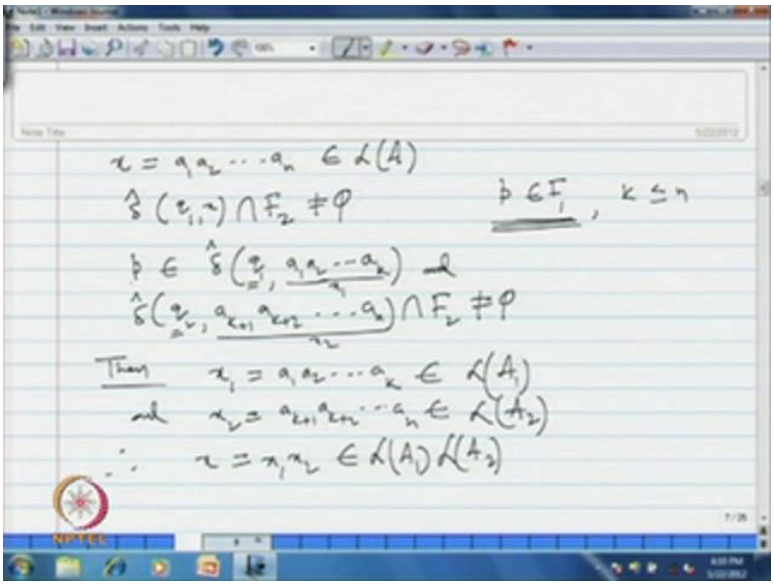
(Refer Slide Time: 22:54)



Formally the construction can be shown to be like this; say A is a ((Refer Time: 23:00)) Q, sigma, delta is q 1 is a star state of A. And, F 2 set of final states of A 2 is a final state of final states of A. So, where Q equal to Q 1 union Q 2, we have not introduce the new states over here. The set of states remain same Q 1 union Q 2 and delta is defined by does define delta like this. So, delta of (q, a) this is basically delta 1 (q, a) we retained all the traditions of this state of this automaton A 1. Therefore, delta (q, a) for this automaton retain all the traditions of this one, accept that there which transition on epsilon from the set of final states 2 the star state of q 2.

So, this is nothing but delta (q, a) equal to delta 1 (q, a) if q belongs to Q 1. If q belongs to Q 1, and A belongs to sigma union epsilon; of course, this will be q 2. If q belongs to F 1 that means; for all states r and F 1 we retained this tradition, whenever it enters a final state of A 1 then, on epsilon it goes to the star state of A 2. Similarly, once it entered the star state of Q 2 we retained all the traditions of A 2. That means; delta (q, a) equal to delta 2 (q, a). If q belongs to Q 2 and A belongs to sigma union epsilon it is as usual. So, these are traditions function is defined for the automaton A that we have already constructed.

(Refer Slide Time: 26:02)



Now, we claimed that L of A equal to L of A 1 continuation L of A 2 to do that suppose that there is string x which is a 1, a 2 up to say a n, and this belongs to say L of A. If this is automaton A that we have constructed from a 1 and a 2 accept the string x; which should have form A 1 up to say n, for each a i belongs to sigma. That is delta hat q 1, x star state of automaton A is q 1, if this process the string x at the star state then, it will eventually reach a final state. That means; this intersection F 2 not equal to phi. This is from the definition of acceptance substituting.

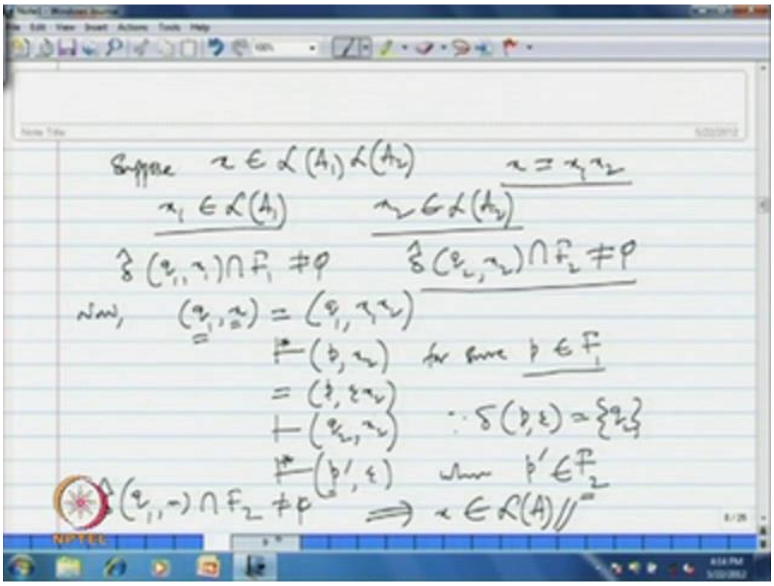
Now, it is clear from the construction of the automaton A; that only way to reach from q 1, any state of F 2 is via this state q 2. Because we have to arrive first one of final states of A 1, from there we have to take an epsilon transition to q 2 and then, only we will be able to arrive at one of the final states of F 1. And, we have only epsilon tradition from F

1 to q 2. Does, it has while traversing through x the automation A from q 1 to some states of F 2. There must exists some state that belongs to F 1, and some number k which is less than or equal to n. Where, n is a numbers of symbols over here.

Such that; p must belong to the set of next states when a process for star state the string a 1 a 2 up to a k sort of processing up to k. The automaton enters the state p where, p is a final state of A 1. From there it has to take epsilon transition to q 2 and then, delta hat q 2, a k plus 1, a k plus 2 up to it has processed the whole string. And, this will eventually reach a final state of a 2 that means; this intersection F 2 not equal to phi. Then, what we found is that the string x 1, which is a 1 a 2 up to a k, this must belong to the language of the automaton A 1. Because it has started a star state of automaton A 1.

And, process the string state is x 1, and p is a final state of F1 therefore, this thing must be accepted this x 1 must be accepting by the automaton A 1. So, this belongs to the language of A 1. And, the x 2 the others string these strings I called it x 2, a k plus 1, a k plus 2, a n. Since, after processing this string x 2 at a star state of q 2, and taking all the moves of a 2 it eventually enters a final state, because it is intersection F 2 is not equal to phi. Therefore, this must belong to the language of A 2. Therefore x, which is equal to x 1, x 2 must belong to L of A 1 concatenation L of A 2.

(Refer Slide Time: 30:39)



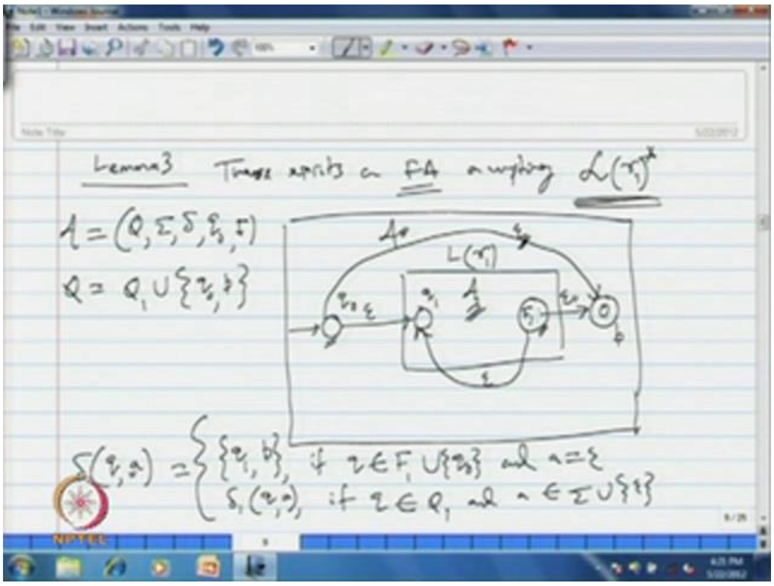
Conversely, we proofed the converse say x sums string x belongs to sigma star and belongs to the language of L of A 1, L of A 2. Then, we can write x as x 1, x 2 such that;

x 1 belongs to L of A 1, and x 2 belongs to L of A 2. So, for some x 1 belongs to sigma star it must belong to L of A 1, and x 2 belong to L of A 2. So, let x 1, x 2 is nothing but x. Therefore, if that is the case then, delta hat q 1 x 1 intersection F 1 not equal to phi. Similarly, from this we get delta hat (q 2, x 2) intersection F 2 not equal to phi, according to the definition acceptance substituting by establishing automations.

Now, if you consider this computation (q 1, x) if q 1 x is this is q 1, x 1 x 2. So, in 0 or more steps eventually it will arrive at p after process the string x 1 and x 2 will yet to be processed. So, where for some p belong to the final states of F 1. Now, from here this concatenation written as p epsilon x 2 and here from this since p belongs to the final state of A 1. We have taken epsilon transition from del state, can go to the star state of a 2 therefore, in one step it will go to star state of q 2, from p epsilon and x 2 will remain, because you know that delta p epsilon occur to q 2.

Now, from this point onward taking 0 or more steps eventually; when is processed it has arrived at some states say p raise, and it has thing epsilon will be x 2 will be exhausted. Where, p raise is a final state of F 2 so, since we have this computation. Therefore, delta hat q 1 x intersection F 2 not equal to phi. So, if you star at q 1 process the whole string x eventually, we arrive at a state p raise which belongs to F 2. Therefore, delta hat q 1 x intersection F not equal to phi. So, this implies x belongs to L of A therefore, we have proved this lemma.

(Refer Slide Time: 34:32)



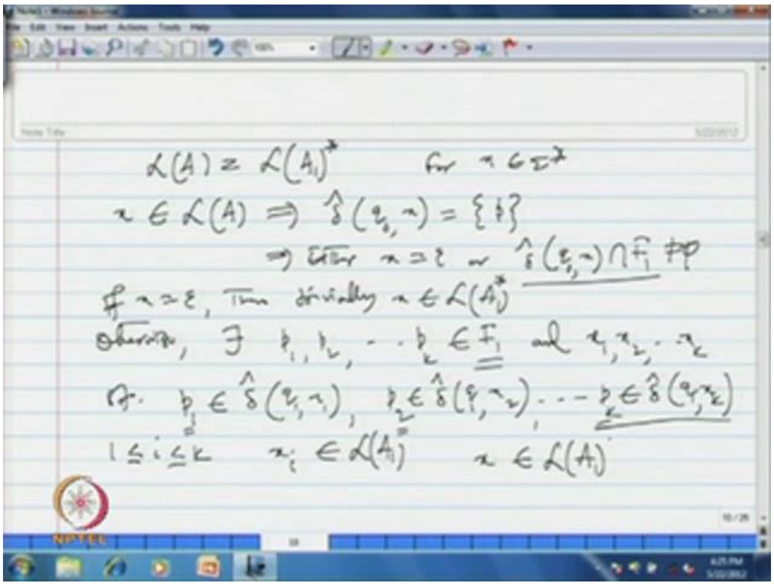
Now, let us show or proof under lemma, which is a lemma 3; this says that given any regular expression r one if we have a finite automaton to accept the language represented by this regular expression r 1. Then, there exists a finite automaton accepting the language L of r 1 star, that means; the clean closer or a regular what a clean closer be regular expression r 1 will have a finite automaton. We can always construct a finite automaton.

So, what we do in the construction it at so, if this is a finite automaton accepting the language represented by the regular expression L of r 1, it will have a star state q 1 and a set of final states set is F 1. So, we construct a new automaton set is A 1 accepting L of r 1. We construct a new automaton where, we introduce a new star state that is q 0, a new final state set is p. And, we consider all the final states of A 1 to be non final states in the automaton A. Then, we give epsilon tradition in front of final states of A 1 to the final state p of automaton A. We provide epsilon transition from final states of A 1 to the star state of A 1. And, also we provide an epsilon tradition from a star state of A to the final states of final state of A.

So, clearly this is a n F A and this n F A; we have constructed from the automaton A 1 for L of r 1. So, clearly the automaton A contains the element Q, sigma, delta, q 0, F where, Q is Q 1 union we have introduced a new star state and final state p. And, delta (q, a) for the automaton A is defined as it will go to state either q 1, or p on epsilon tradition. If dusted q belongs to the final state of so this epsilon tradition, if you belongs to this final state of A 1. Or, it is a star state of A; that means; for this epsilon transition. So, if Q belongs to F 1 union q 0 then, on epsilon tradition A equal to epsilon, it will go to either q 1 F 1 to a q 0 to sorry, this is q 0 to q 1, or F 1 to p or Q 0 to p. And, then it takes or retains all the traditions of the automaton A 1, if q belongs to q 1 and A belongs to sigma union epsilon.

So, that is how we have constructed the automaton A, from the automaton A 1 and you claimed that this automaton A accepts the language L of r 1 star. This because without taking an input at a star state; it may go to the star state of A 1. And then, using this transition from the final states of once it reaches the final state using this transition by again come back to the star state of q 1. And, it can be done many numbers of times so, that denotes r 1 star. Because this automaton A 1 accepts L of r 1 so since; we have introduced this loop on epsilon transition. From the final state to the star state from the final state of A 1 to the star state of A 1. Because of this anything that; lead the automaton A 1 from the star state of final state that can be taken many numbers of times. And since; because of this epsilon transition from star state of q 0 to the final state of star state of a q 0, to the final state p of k, epsilon is also accepted by the automaton A. So, therefore this is nothing but L of r 1star, let us proof this formally.

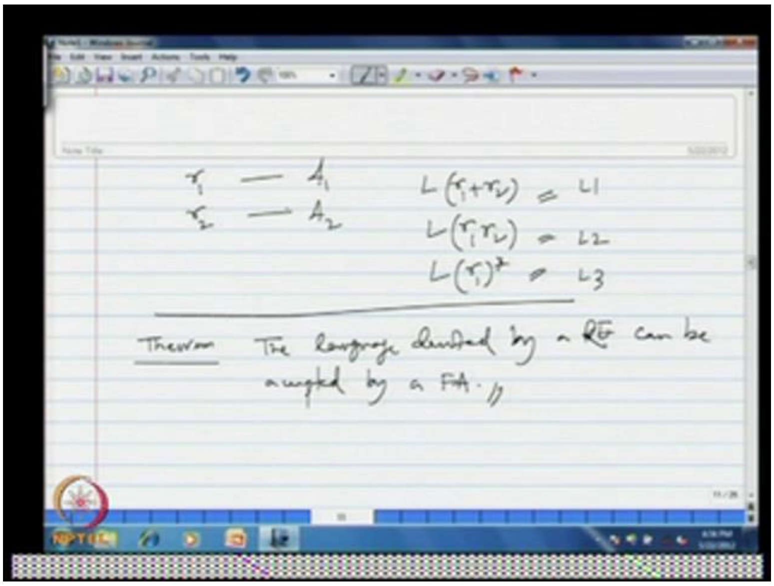
(Refer Slide Time: 41:41)



We proved that L of A is nothing but L of A 1 star. Now, for x belong to sigma star let us consider any string our sigma star x belongs to L of A means; delta hat (q 0, x) for star you possess this eventually it must arrive at state p. Where the final state of a automaton A. This implies that either x is epsilon in subs case from star state q 0, directly we can go to state p. Or, delta hat (q 0, x) intersection F 1 not equal to phi, that means; we process the string x at state q 0 eventually arrive at on a final state of A 1. And, from here we can take the epsilon transition to p; that we can accept the string. So, either x equal to epsilon or this must be true.

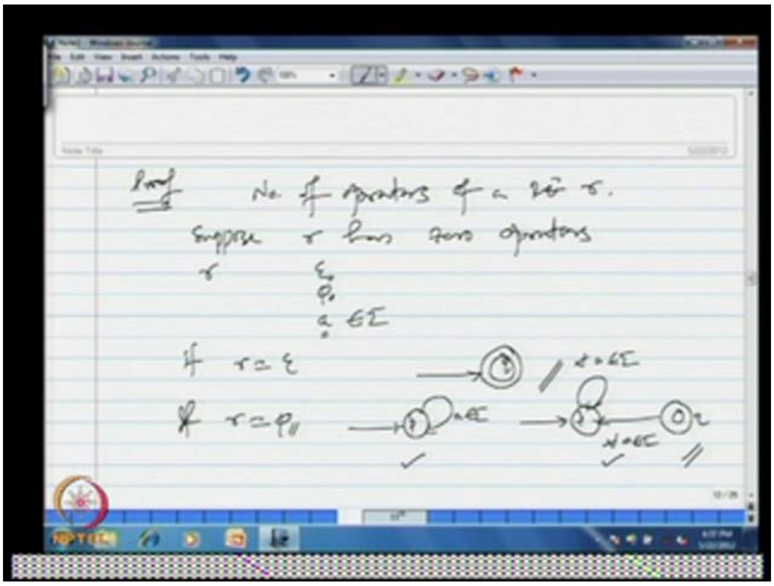
So, if x equal to epsilon then, trivially x belongs to L of A 1 star, so it is quite trivial. Otherwise, there exists a sequence of state say p 1, p 2, p k which belongs to F 1 and some substring of x is x 1, x 2 up to say x k. Such that; we can arrive at p 1 after processing the string x 1 at a star state q 1 process the string x 1 at q 1. This extension function then, we can arrive at string p 2 that means; p 2 belongs to delta hat q 2, x 2 and so on. Eventually; p k belongs to delta hat sorry, this is q 1, I have written as star at star state of the automaton A 1. So, q 1, x k process the string x k at a star state of q 1, and eventually; it leads us to state p k. Thus for all i greater than or equal to 1 less than or equal to k, x i must belongs to L of A 1. Because each case p 1, p 2, p k it belongs to state set of a final states. Therefore, it is substring x I, x 1, x 2, x k must belong to the language L of A 1. Therefore, x belongs to L of A 1 star.

(Refer Slide Time: 46:01)



So, we have shown a proofed 3 lemmas that means; given automaton finite automaton for the language represented by r 1 that is A 1. And, for r 2 if the automaton is A 2 then, we can always construct finite automaton accepting the language is L of r 1 plus r 2, L of r 1 r 2 and L of r 1 star. Now, we are going to proof the theorem; that the language denoted by a regular expression can be accepted by a finite automaton. So, in that we will be using these 3 lemmas say; lemma 1, lemma 2 and lemma 3. Now, let us proof this theorem.

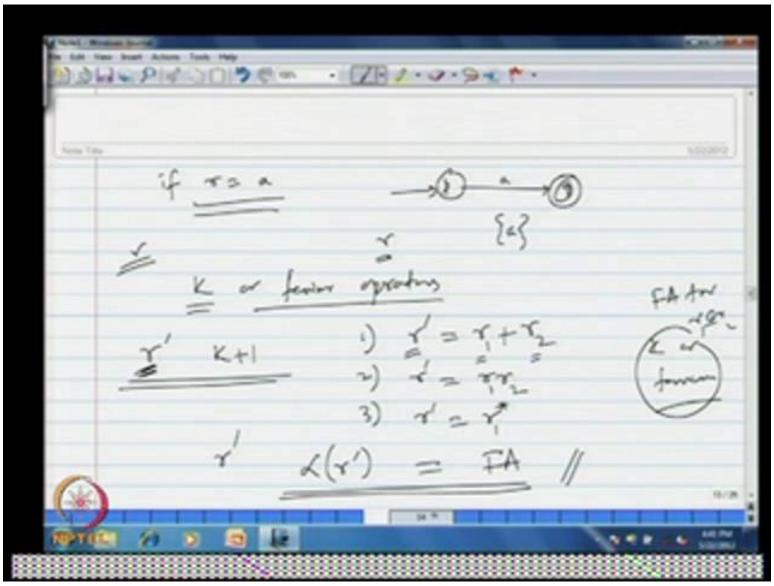
(Refer Slide Time: 47:02)



So, proof of that theorem. So, we have proofed that result by induction on the number of operator away regular expression. So, number of operators of a regular expression r, we apply index on that. Suppose, r has 0 operators then, that is the basis case in subs case r must be either epsilon, or phi, or can be a single symbol a, that belongs to the input alphabet. Because there is no operators involved, it must be single symbol, it maybe phi, or it may be epsilon. For each of the cases; we can construct finite automaton to accept this. For example; if r equal to epsilon then, the finite automaton containing the single state and which is a star state and also final state.

So, this state will accept this finite automaton will accept the string epsilon. Similarly, if r equal to phi then, a finite automaton of this form said it has state p star state, and for any symbol a belonging to sigma. There is self look and there is no final state in such a case no sting will accepted by this finite automaton; and hence r equal to phi. Or, we can also do it like this, we can incorporate a final state p just a star state. So, for all a belonging to sigma; we can self look here. And then, this is the final state say it is q and we give a transition from q on all a belonging to sigma. So, since there is no path from star state to the final state this will also accept empty set. Therefore, there is a automaton either this or this one is automaton to accept the empty set.

(Refer Slide Time: 49:44)



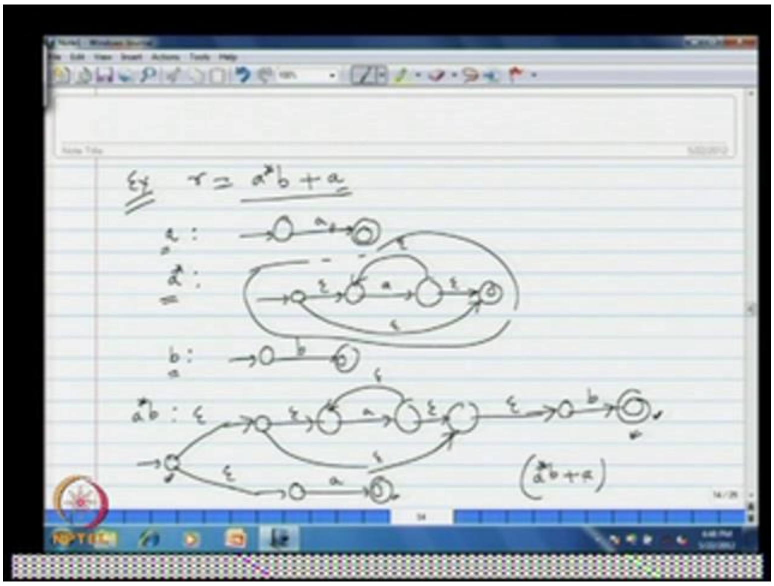
And then, if r equal to single symbol a then, this automaton with star state p on a it goes to the final state q. So, this automaton will accept the language of the automaton will be simply the single term a. Therefore, this automaton accepts r, where r equal to a. Now suppose that the reason is true for regular expressions with say k or fewer operators. Suppose, r is a regular expression, which has k or fewer operators and for that case; assumed that the result is true. We have the finite automaton to accept the same language represented by r; that is the ((Refer Time: 50:47)).

Now, consider a regular expression r that has k plus 1 operator. When to show that for this regular expression also or say r dash which has k plus 1 operators then, for this regular expression also we will able to construct a finite automaton. So, how to do that if this expression regular expression has k plus 1 operators then, there 3 cases according to the operators involved in regular expression, or with regular expression. So, the number 1 this r s must be r 1 plus r 2.

Number 2; it maybe this r s must r 1 r 2, or number 3; this r s must be some r 1 star. So, we have considered these some concatenation and clean closer. So, in any case you note that both r 1 and r 2 since; r s and k plus 1 operators both r 1 and r 2 must have k or fewer operators. Therefore, for each of this r 1 and r 2 we have already finite automaton for r 1 and r 2 is already available, according to the in that ((Refer Time: 52:26)).

Now, we have already shown by using this lemma 1, and lemma 2 and lemma 3; that if there is the finite automaton for r 1 and r 2. Then, we have finite automaton for L of r 1 plus r 2, L of r 1 r 2 and L of r 1 star. Therefore, for r s which has k plus 1 operators so, this is nothing but r 1 plus r 2 or it may be r 1 r 2 or r 1 star will have finite automaton accepting r s. Therefore, for L of r s we have an finite automaton therefore, given any regular expression r we able to construct a finite automaton, accepting the language represented by this regular expression. So, this compliance that proof the theorem; that for any given regular expression we can construct a finite automaton accepting the same language.

(Refer Slide Time: 53:46)



Now, let us see an example; demonstrating the construction of the n F a for a regular expression. Consider regular expression as r which is a star b plus a, we will just follow the steps which we have already described to construct finite automaton for regular expression r. First, we lease the corresponding n F a for each sub expression of a star b plus a. For this a and the first sub expression the corresponding automaton according to our construction is this one, containing a symbol 2 states star, and final state with a single transition on symbol a. So, this automaton accepts the regular expression simply, from this where the construction of clean closer.

We can have for a sub expression a star we can construct the automaton like this. We start with this automaton this is we introduce a new star state, and the final state and we make this to be non final state. Then, these are new star state given epsilon transition directly to this final state epsilon transition to this star state of the previous automaton. Epsilon transition to the final state from the previous final state, and from this final state of the previous automaton to the star state we give an epsilon transition. So, according construction this automaton will accept the language of the regular expression is a star. Then, for b similarly; expression b we have the automaton containing two states, where there is single transition on input symbol b. Therefore, for a star b that is concatenation of this a star and b. We can now construct the automaton like this first you consider this automaton and then, epsilon transition to this one.

So, these are automaton for a star, and these are automaton for single term b. Now, according to the construction we consider this to be a non final state. And, from this non final state we give an epsilon transition to the star state of this automaton. And, we make these to be the star state of the automaton, and these are final state of this automaton. So, this will accept the language the regular expression a star b. Therefore, the automaton for a star b plus a will be, this automaton union we will create now, a new star state and the automaton for a will be. For this a we draw here, a transition diagram for the automaton with the single term a, and you create a new star state give epsilon transition to the star state of previous automaton and epsilon transition to star state of this automaton. So, these are star state of the new automaton and this one, and this one will be the final state of this new automaton, which will accept the language a star b or a.