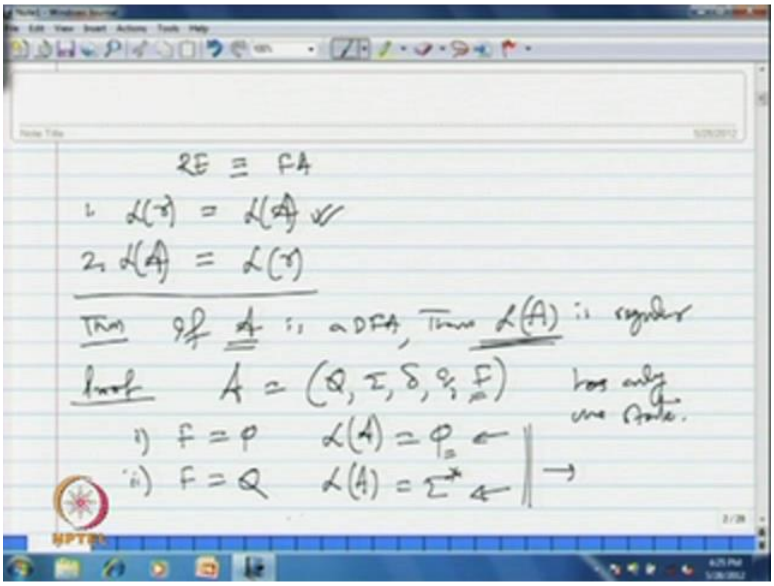
Formal Languages and Automata Theory
Prof. Diganta Goswami
Department of Computer and Engineering
Indian Institute of Technology, Guwahati

Module - 5
RL – RG - FA
Lecture - 2
RE – FA

(Refer slide Time: 00:22)



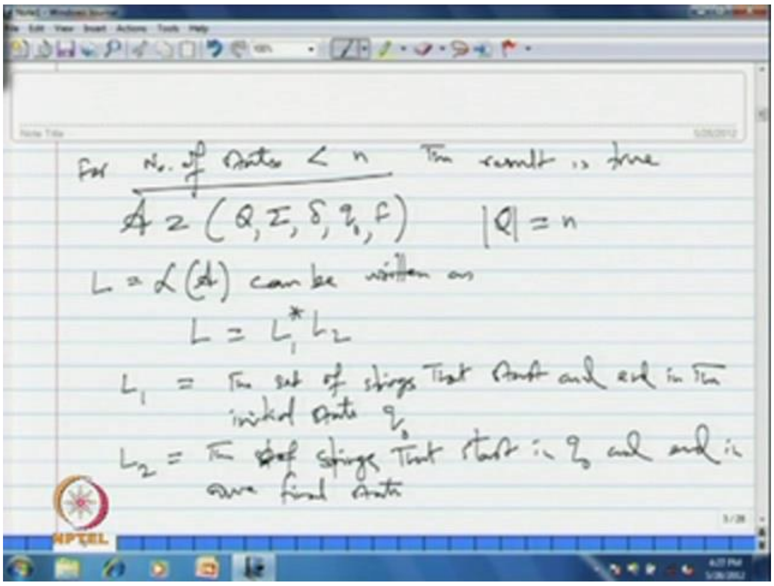
We want to show that regular expression and finite automata are equivalent. We have already said that to prove this, we need to prove two points. First is that given any regular expression r, we should able to construct a finite automaton A, so that the language accept by A is precisely the language represented by the regular expression. And then given any D F A which should able to construct a regular expression equivalent regular expression, that means the language accept by the finite automaton A is exactly the language represented by r.

So, we have already shown the first step in the last lecture. For any given regular expression we have, we constructed an equivalent finite automata A to introduce this lecture we are going to show the next step, the second step. That means; we are going to prove this theorem, if A is the D F A then L of A is regular. That means; there exists the regular expression r representing the same language L (A) accepted by the D F A. So, we will prove this by enacts some numbers of states with D F A, consider for base case; the

D F A to say A, D F A containing the set of states Q alphabet, sigma, delta is a transition map q naught is the start state and, F is the set of states.

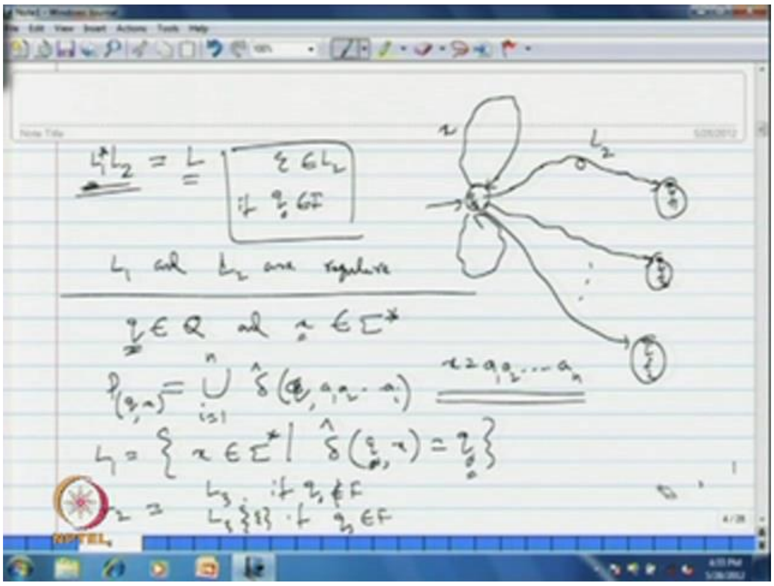
So, for a base case let us consider that this D F A has on the 1 state only 1 state so, it is the base case. So, in this case there are 2 possibilities for a set of final states F; that this D F A has. First F equal to phi that means; there is no final states, in such a case L of A the language accepted by A is nothing but the empty language. But this phi is a regular expression, and hence this is regular. And then the other possibility is that F is set of states that means; only state that the D F A has is a final state. In such a case L (A) is nothing but sigma star, it accepts all strings over sigma. But sigma star is also a regular and, hence in both caseses we have seen that this is there is a regular expression representing the language accepted by the finite automaton A.

(Refer Slide Time: 03:31)



Now, let us consider that the result is true for all those D F A, whose numbers of states is less than n. So, numbers of states less than n the result is true. That means; for a d F day D F A we can construct any regular expression, that means; the language accepts by the D F A is a regular. So, lets the D F A to be A containing the elements Q, sigma, delta, q 0, F, say numbers of states now is say n. So, first note that; the language L accepted by the D F A can be written as L equal to L 1 star L 2. So, where L 1 is the set of strings that start and end, the initial state q 0. And then L 2 is a set of strings that start in q 0, the start state and end in some final states.

(Refer Slide Time: 05:43)



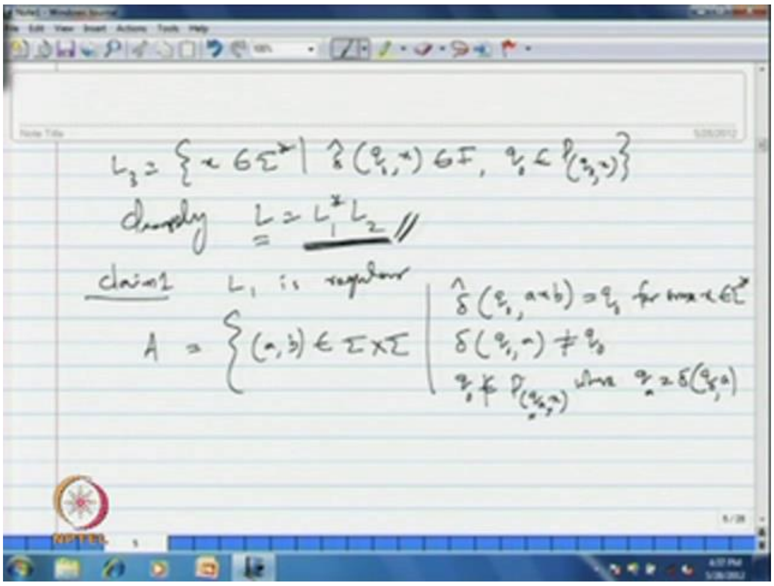
That means; we can consider that say this is the start state q 0. So, from q 0, if a string leads us from q 0 to again q 0, say this is after process the string q 0 on the state q 0. If it brings us back to the same state q 0 and then suppose for some other strings it again brings us to the same state q 0 then union of all those strings, if we consider as L 1. And, we can take this particular string n numbers of time, because you can search q 0 come back to q 0 and continue again with that string and come back to q 0. We can take n numbers of times that is why it is L 1 star. And then other consider strings is that start state q 0, and process the string, eventually we arrive at some final state say q F 1. Similarly, you can process some other string and arrive at state q 0, and arrive at some other final state q F 2 and so on, say q F k.

So, consider the union of all those set of strings. So, first we take the strings of this kind from this language and then we take consider this language L 2. Union of all those strings which lead us from q 0 to some final state q F, q F 1, where q 0 is not an entrance state over here, in this path. So, we can take this string n numbers of times and then concatenate to it a string from L 2, and that will be the precise that will be precisely the language L accept by this D F A. We include epsilon in L 2, if q 0 is your final state; if q 0 is final state then epsilon belongs to L 2 we consider this situation.

Now, using the inductive hypothesis; we proved that both L 1 and L 2 are regular. That means; since L 1 and L 2 are regular, we have constructed L by using only the regular operation L 1 star concatenation L 2. Therefore, L must also be regular. Now, we will be using the notations for defining language L 1 and L 2. Suppose, q belongs to the set of states and exceeding over sigma star now, we will denote the set of states on the path of x from q that come after q. That means; once you process the string x at state q then we will be will arrive at some set of states. And, those set of states is basically denoted as p q x. So, if you consider x to be a 1, a 2 up to suppose some a n so, after processing the string a 1, a 2 up to i then we will arrive at some state starting with i equal to 1 up to n. We take a union of all those states that is the set of states that we will reach. If you process the string x at state q, and you denote this to be p (q, x). So, p (q, x) is a set of states that we arrived at by processing the string x at some state q.

Now, we defined L 1 to be the set of strings x belong to sigma star. Such that; delta hat (q 0, x) equal to q 0. So, we have defined L 1 to be the set of all strings x such that; processing x at initial state q 0 eventually it has to the same state q 0, by this self flow. Similarly, L 2 is basically L 3; if q 0 does not belong to F, and it is equivalent L 3 union epsilon. If q 0 belongs to F, where L 3 basically it is a set of strings x belongs to sigma star. Such that; delta hat(q 0, x) belongs to F so, if you write it.

(Refer slide Time: 11:54)

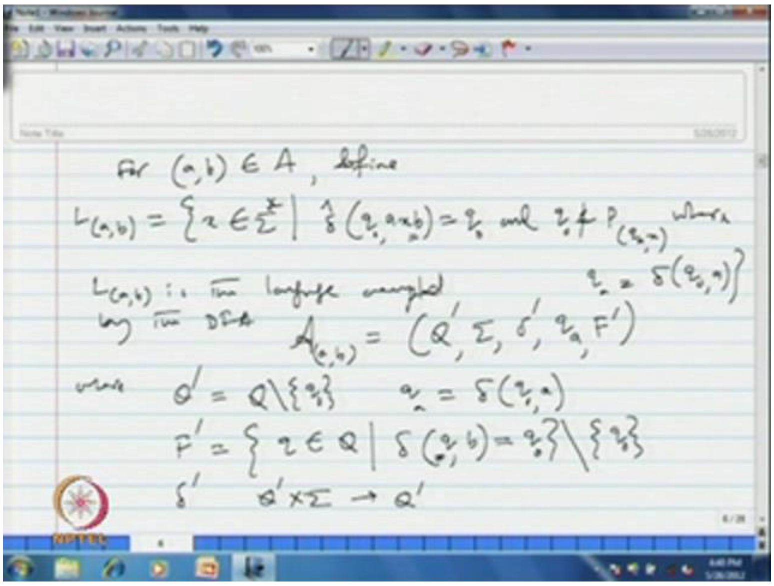


So, where L 3 is the set of strings x belong to sigma star. Such that; delta hat (q 0, x) belongs to the set of final state. That means; if process the string at the initial state q 0, we eventually reaches one of the final state F. And then q 0 does not come in the part while processing x that means; q 0 does not belong to p (q 0, x) according to our notation that we have just defined. So, clearly L equal to L 1 star L 2, because you can take the strings of this form L 1 from L 1 n number of times. And, if we concatenate with L 2 eventually we reach or arrive at one of the final state of the D F A. Therefore, this any string of this form will be accepted by the D F A a, and hence is a language precisely this is a language accepted by the D F A.

Now, we will first prove that; L 1 is regular. That means; there is regular expression for a language L 1 and then we will prove that L 2 is regular. If we can proved that L 1 and L 2 both are regular then L must also be regular, because we have constructed L by using only regular operations. So, to proved that L 1 is regular; we consider the set, say A is a set, the set of pairs of this form say (a, b) where, a b a and b or symbols from sigma. That means; it belongs to sigma cross sigma. Such that; delta hat q 0, a x b equal to q 0, for some string x belong to sigma star. And then delta (q 0, a) is not equal to q 0 that means; if we process a taking single symbol A at q 0 it does not come back to the q 0. It will go to some other states and then q 0 does not belong to the set of states p(q 0, x). That means; it does not come in the path while processing x at state q 0.

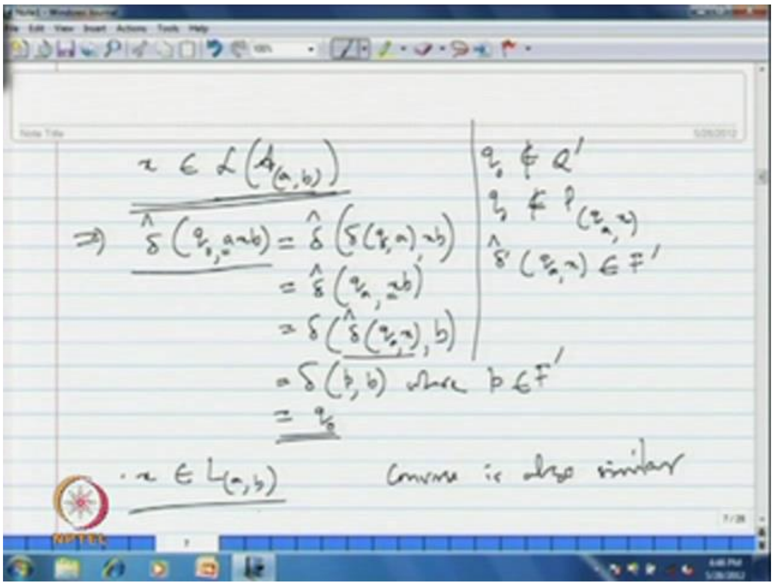
So, this is q q 0 does not belong to (q a, x) that means; q 0 does not come in the path while process x at state q a. Where, (q, a) is nothing but; the state that we get while taking the input symbol A at state q 0. Since, we have already said that; delta (q 0, a) is not equal to q 0, it must be other state and it say q a, and while processing the string x at state q a, q 0 does not come again in the path.

(Refer Slide Time: 15:36)



Now, for this pair (a, b) belong to the set A. We defined the language L (a, b) for the pair a b we defined the language a b, L a b is a set of all the strings at x belong to sigma star. Such that; delta hat q 0, a x b equal to q 0, and q 0 does not come in the path p (q a, x) where, q a is nothing but delta q 0, a. so, we will define the language L(a, b) like this. Now, we will show that L (a, b) this language we have just defined is the language accepted by the following D F A, say A (a, b) D F A we define it like this. This Q dash, sigma, delta dash, q a is a start state and F dash. So, where q dash is nothing but the set of previous state accept for the start state q 0. So, we leave or start state from original set of states, and there is a set of states of the new D F A, A (a, b). And, q a the start state of this D F A is nothing but the state that we arrived at by taking the input symbol a at state q 0. And, the set of final state F dash, it is all those states belong to the state q. Such that; delta (q, b) that means, when you process eventually this last symbol b whenever, it goes to q 0 then all those states q for which taking symbol b it goes to state q 0, it is consider to be a final state. Accept for of course, the state q 0, q 0 is not in the set of states. And, delta dash we retain the same set of transition functions with the restriction that; it is from Q dash cross sigma to Q dash.

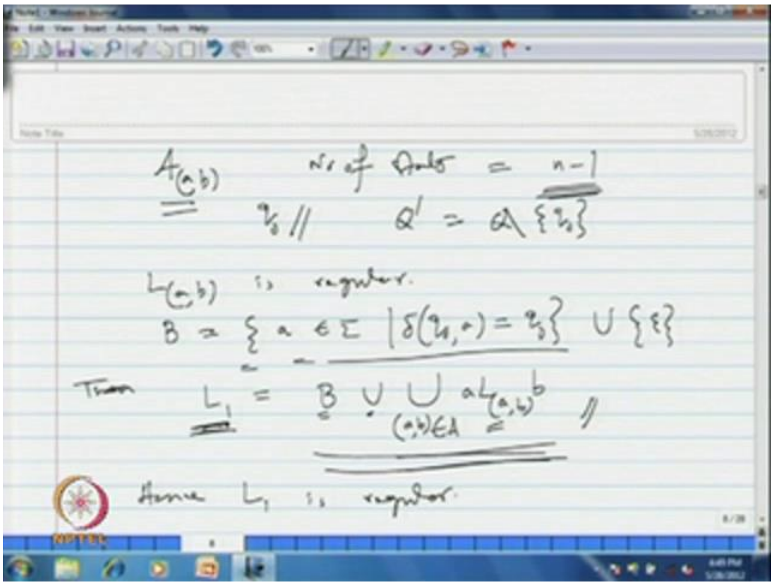
(Refer Slide Time: 19:02)



Now, suppose the string x belongs to L of (a, b) so clearly; a b is a D F A. Suppose the string x belongs to L of (a, b) now, since q 0 does not belong to the set of states in this new D F A (a, b). And, q 0 also does not belong to the set of states that you can arrive it by processing. While process the string starting at q a, the start state and delta dash hat (q a, x) belongs to F dash, whenever say x belongs to this language of this D F A. Therefore, this implies that delta hat q 0, a x b is nothing but delta hat delta (q, a) x b, that means; first process symbol a at state start state q 0. And, then compute the process the string x b at that state, concentrating this accentuation function. Therefore, this can be written as delta hat since; delta (q 0, a) is nothing but q a so, this is x b. But this can be written as delta, delta hat (q a, x) so, first process the string x. Therefore, we used accentuation function and then compute delta of wherever we accept that state, we compute delta of that state b. So, this is nothing but delta of some p where, p is the state that we arrived at after processing this string x at state q a.

So delta (p, b) where, p must belong to the set of final states, because this string accepted by this D F A. And, hence p must belong to some final state. According to our definition this is delta (p, b) is q 0 so that since delta hat q 0 a x b equal to q 0. According to our definition of the language L (a, b) we know that this is not this string x belongs to L (a, b) the converse is also similar. That means; converse is also similar, we can prove it similarly, that means; if this thing x belongs to L (a, b) this x will be accept by the D F A L D F A (a, b).

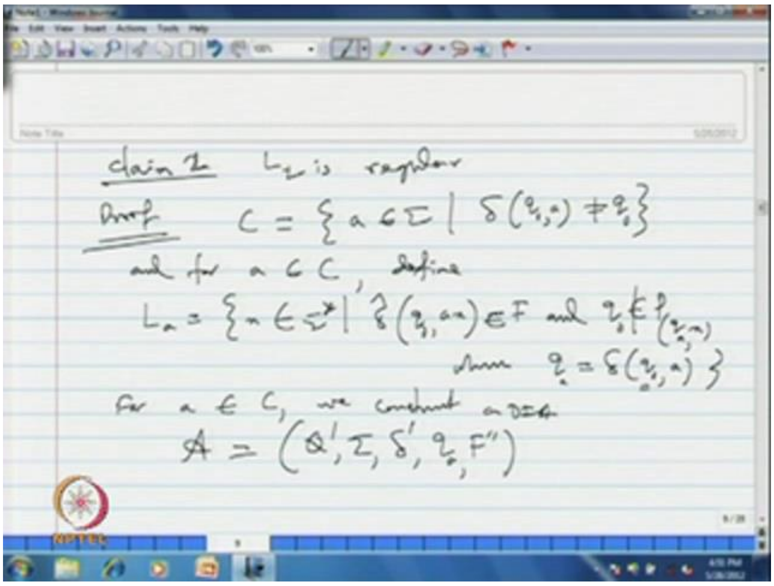
(Refer Slide Time: 22:50)



Hence, what we have done is that; in the D F A A (a, b) the numbers of states we have is exactly n minus 1, because we have already omitted the start state q 0 from the set of states. That means; Q dash counting’s all states in Q accept for q 0, and the numbers of states in the D F A A (a, b) is exactly n minus 1. Therefore, according to induction hypothesis the language L (a, b) is regular.

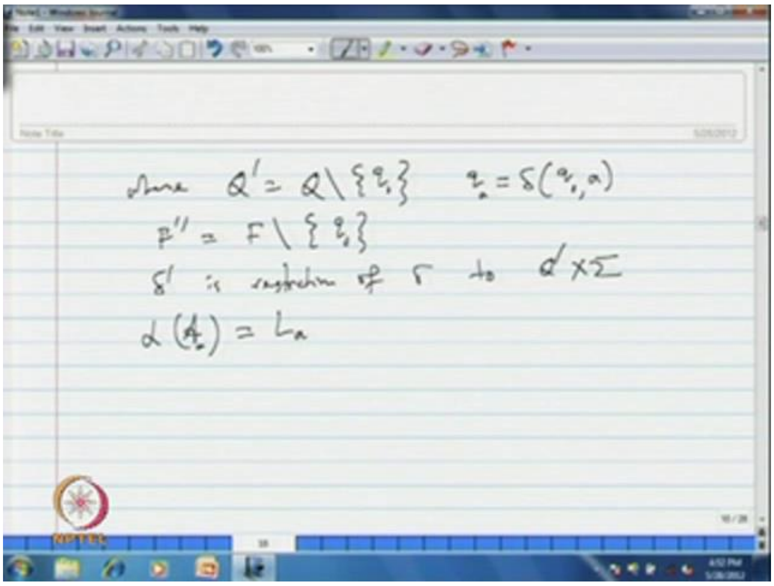
Now, if we write the set B to be all symbols a belong to sigma, such that; delta (q 0, a) equal to q 0, union the string epsilon. Then, clearly you can see that the language L 1 that we have already defined is nothing but this B union set of all strings of this form a L(a, b) then b. Where, this pair (a, b) belongs to the set A, and this language L 1 concatenate by taking union of a regular set this set is regular, set B is regular. We have shown L (a, b) to be the regular, we have concatenated with a and b so, a concatenation L (a, b) and I have taken the union of those regular languages. Therefore, L 1 must be regular since; we constructed L 1 by using some regular operations over some regular languages. Therefore, L 1 is regular hence, we have seen that L 1 is regular.

(Refer Slide Time: 25:04)



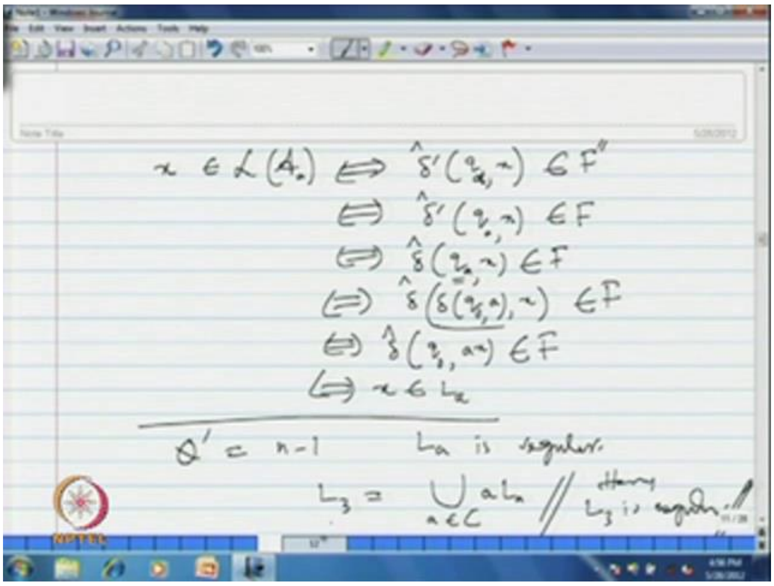
Now, in order to prove that the language L 2 is also regular now, the proof this claim 2 that L 2 is regular, we consider set say C, the set of all symbols a belong to sigma, such that; delta (q 0, a) not equal to q 0. So, once you process or take a symbol a at start state q 0, it does not go to q 0 again. And, for a symbol a belong to C; we defined the language L a to be a set of all strings x belong to sigma star such that; delta hat q 0 a x start with a symbol a and process the string x. That eventually leads us to some final state, and q 0 does not appear on the path while processing it from death state q a, where q a is exactly death state. That we arrived at taken a symbol a at the start state q 0. Now, for symbol a belongs to set C; we construct a D F A, A which is exactly Q dash sigma delta dash q a and F double dash.

(Refer Slide Time: 27:20)



Where, Q dash is the set of state Q accept for the start state q 0, that means; this D F A a contains called it D F A to be say A this D F A contains 1 state less than the previous D F A. And, then q a is the start state of this D F A a a is exactly delta (q 0, a) that is it that we have arrived at by taking symbol a on state q 0. And, the set of 1 state F double dash is nothing but all those states final states of previous D F A accept for the start state q 0. And, delta dash is again the restriction of delta to delta dash is basically a restriction of delta to Q dash cross sigma. Now, this is observed that L of A the language accepted by the D F A a a is exactly L a. Now, first note that q 0 does not appear in the context of L a and L a a.

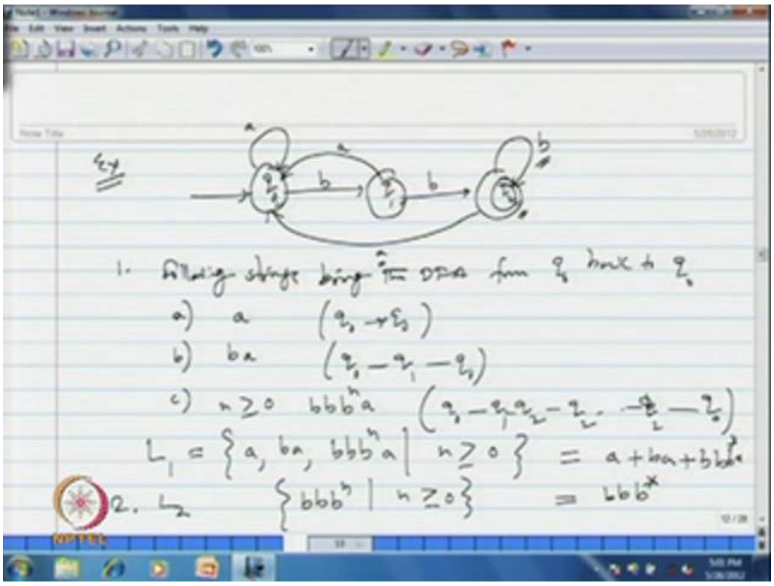
(Refer Slide Time: 28:46)



Now, if x any string x belongs to sigma star belongs to L of a language of this D F A a a. So, this belongs to this language of D F A, if and only if; delta dash hat q 0 sorry, q a x because the start state this q a process the string x. Eventually belongs to some final state of this D F A. Since; q 0 does not appear this if and only if delta dash hat (q a, x) belongs to F. Since, we have written all the final states of the previous D F A. This if and only if delta hat, because we have written the same transition function restricted to the new set of states, we can replace delta dash by delta. If this belongs to F if that is the case; we can write it like this it is delta dash q g. So, appending a we are taking a the beginning if we start at q 0, take a since; q a is nothing but (q 0, a). If we take delta we can write as delta q 0 a x if this belongs to F. So, this is simply this q a is written as; delta q 0 a this if and only if delta hat, we can write it as delta. We can write it as, since this delta q 0, a x we can write it as delta hat q 0, a x accentuation function process the substring process string a x if this belongs to F. Since, q 0 does not appear.

So, this implies that according to our definition of the language L a; we know that this is nothing but x belongs to L of a. We have shown that; if x belongs to x belongs to L of a a if and only if x belongs to L a. So, again the numbers of states in Q dash is exactly n minus 1, because we have omitted the start state q 0 in a A a. Therefore, by inductive hypothesis; the language L a is regular. But clearly the language L 3 is nothing but; union of those languages a L a thus concatenate a with L a. And, for all of a belong to the set C. Therefore, L 3 is also regular hence; L 3 is regular hence this completes the proof of the theorem.

(Refer Slide Time: 32:31)

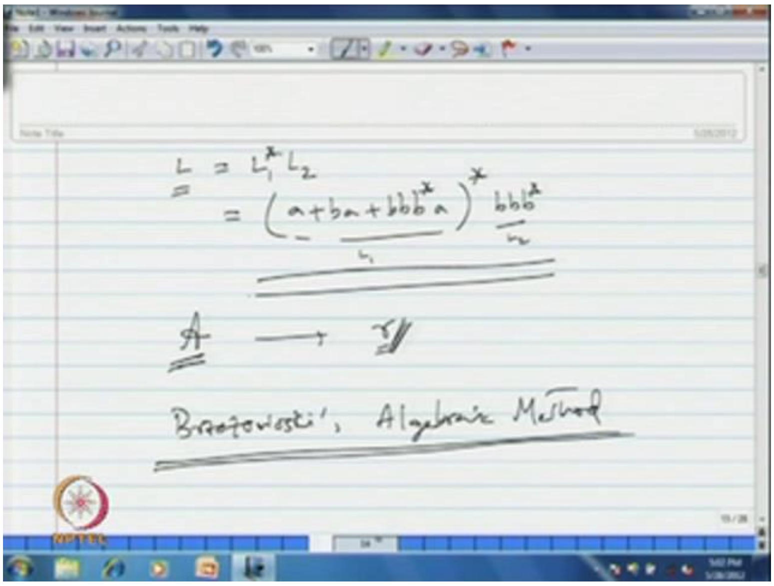


That means; given any D F A the corresponding language accepted by the D F A is regular. That means; we can always construct a regular expression for the language accepted by the D F A. Now, let us just given an example; consider the D F A given by the transition function here, say q 0 is the start state. On a remains at q 0, on symbol b it goes to state q 1, q 1 on a it comes to state q 0, on b it goes to q 2, where q 2 is the final state, q 2 on a comes back to state q 0, and on b it remains the same that q 2, consider this D F A.

Now, you note that the following strings bring the D F A from q 0 back to q 0. That means; do there string that will be there in the language L 1 for this particular D F A. For a means via the path q 0, q 0 because you can take a self look start at q 0 on a, it will back to q 0. Then, again if we take that this path from q 0 to q 1 then again back to q 0 taking a we will get the string b a. If we take the path start at q 0 go to q 1, and back to q 0 and then we can also construct the path. We can from q 0 you go to q 1 on b, from q 1 you go to q 2 on b and on any numbers of b you remain at q 2. And, eventually on taking following this path taking a and you come back to q 0. That means; for n greater than or equal to 0; the strings of the form b b b to the power n a.

So, all those strings will bring us back from q 0 to q 0 again. If we take the path q 0 q 1 q 2 then you remain at q 2 for as many times as we want by taking a b. And, eventually by taking an a you come back to q 0 so, this is 1 possibility. Thus L 1 can be written as string a, or b a, or b b b to the power n a, for n greater than equal to 0. Again since; q 0 is not a final state, therefore; L 2 the set of strings which take the D F A from q 0 to the final state q 2. There is only 1 final state from the start state q 0 what kinds of string brings us from q 0 to the final state q 2. Where, q 0 is not in the path it will have the form wherever take this b, taking b we will have to go to q 0. And, from q 1 we can taking b we can go to q 2, and we can take n numbers of time this b using a self look. Therefore, L 2 is basically of the form b b b to the power n where, n is greater than equal to 0. Now, L 1 is of the form, or L 1 can be written as or expressed by the regular expression a or b a or b b b star a, and L 2 can be written as b b b star.

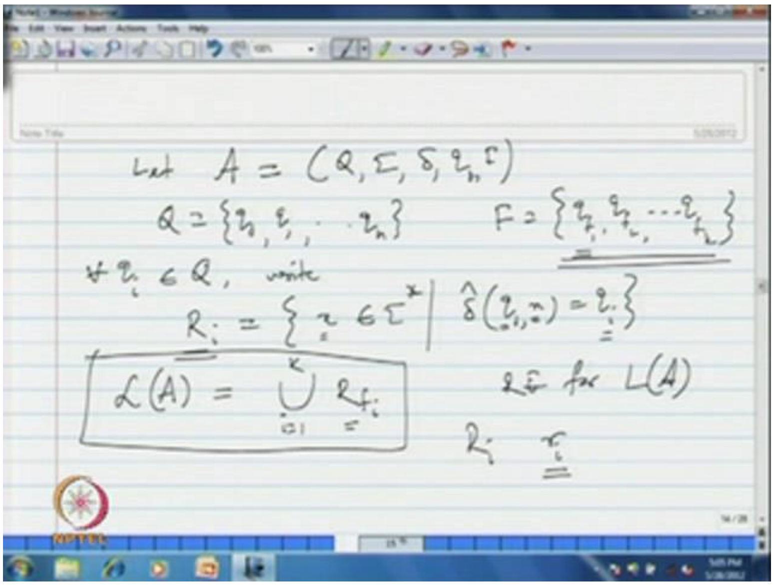
(Refer Slide Time: 37:05)



Thus, let us for the construct above theorem; the language accepted by D F A is basically L which is L 1 star L 2. That means; this can be written as or expressed by the D F A the expression a or b a or b b b star a star this is L 1, this star L 2, L 2 is basically b b b star. So, this is the regular expression corresponding to this L 2 so, L 1 star L 2. So, these are regular expression corresponding to language L accepts by the D F A. Now, what we will do is that; the main point is that given any D F A which should be able to construct a equivalent a regular expression. And, we can always construct equivalent regular expression in the sense that; the language represented by this regular expression r is exactly the language accept by the D F A.

Now, how to construct a regular expression r for equivalent regular expression r for any given D F A, so here is the algebraic method given by the brzozowski that means; brzozowski so that is algebraic method which has proposed by brzozowski. We will just now, we will now discuss this brzozowskis algebraic method for construction the regular expression for any given D F A.

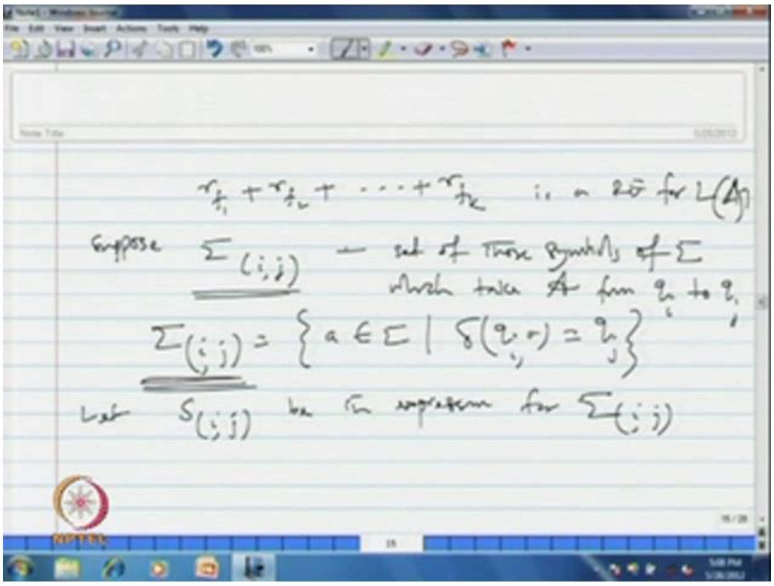
(Refer Slide Time: 38:46)



Let, A be your D F A containing a triples represented by the triple ((Refer Time: 38:58)) sigma, delta, q 0, F. Where, the set of states is basically q 0, q 1 up to say q n, n numbers of state. And, the set of final states we have said k numbers of final states say q f 1, q f 2 up to say q f 2. Now, for every i, every state q i belong to the set of states q, right. Say R i is the set of strings that means the language, or say set of strings belong to sigma star. Such that; delta hat (q 0, x) equal to q i, that means; we process the string x at any state that q 0 process the string x at any state q 0. Eventually, we arrive at a state q i we collect all those strings x to construct the set R i.

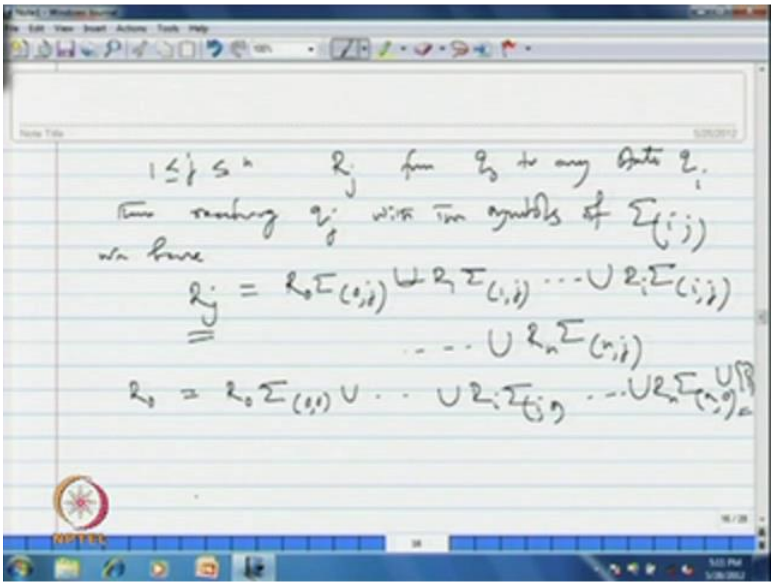
Now, we will note that the language of this D F A is nothing but union of all those sets R f i. Where, i equal to 1 to k, because if we consider for every state q f 1 the corresponding set R f 1, R f 2 and R f k. And, if we take all those strings take the union of all those strings that is exactly the language of the D F A. Hence, L (A) can be written as the language of the D F A can be written as union of all those sets r F I, for i equal to 1 to k. Now, in order to construct a regular expression for L (A) for the language of A, we propose an unknown for each r i say it is r i. So, for each R i we propose an unknown say regular expression say r i.

(Refer Slide Time: 41:26)



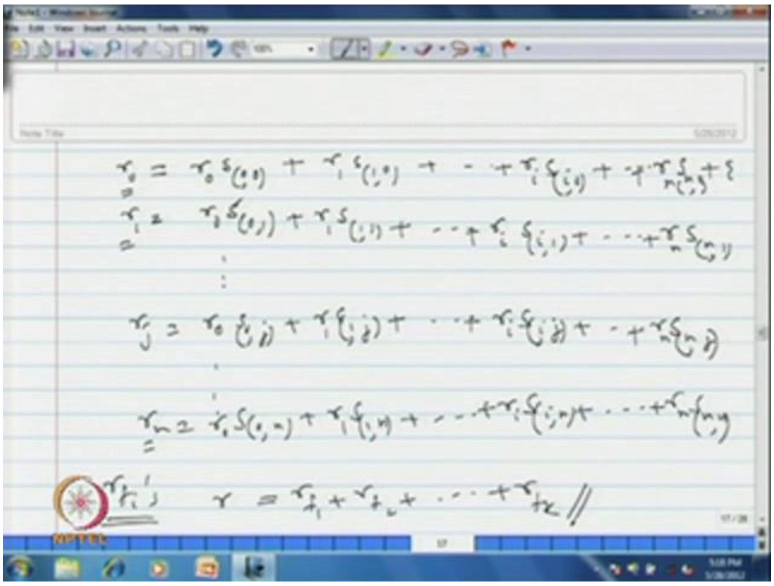
We observed that; r f 1 plus r f 2 up to say r f k is exactly is a regular expression for the language of the D F A L (A). Suppose, say sigma (i, j) subscript (i, j) this is a set of those symbols of sigma, which take the D F A from the state q i to the state q j. We just assume the sigma (i, j) this notation I used for a set of those symbols of sigma which take that D F A a from the state q i to the state q j. That means; sigma (i, j) is nothing but all those symbols a belong to sigma such that; delta q i a equal to q j that is what exactly we have defined. Clearly as it is a finite set sigma (i, j) is regular with the regular expression as saw with symbols. Whatever the symbols you had wrote here, if we submit up that with regular expression for sigma (i, j) it always finite. And, you can write a regular expression for it. Now, let s (i, j) s subscript (i, j) be the expression that expression for sigma (i, j) so whatever, the regular expression we get say denoted by s (i, j).

(Refer slide Time: 43:56)



Now, for say j greater than equal to 1 less than equal to n since; the strings of R j are precisely taking the D F A, from q 0 to any state q i then reaching q j with symbols of or the symbols of sigma (i, j). We have R j to be we can write it as; R 0 sigma 0 j so, R j with a set of strings that takes us from q 0 to q 0. And, then just consider 1 symbol is which we will take from q 0 to j q j denoted by sigma 0 j union R 1 sigma (1, j) set of all those strings which takes the D F A from q 0 to q 1 and from q 1 to q j on a single symbol from of j sigma (1, j). Like that union R i sigma (i, j) and eventually R n sigma (n, j). So, any R j can be written as a union of all those, and in the case of R 0 it is basically R 0 sigma (0, 0) union R i sigma (i, 0) to say R n sigma (n, 0) union of course we have this string I have the string epsilon as epsilon takes that D F A from q 0 to itself without taking any state, that is how you have taken epsilon at the end. Thus for each j we have the equation for R j which depending on all R i s called a characteristic on R j.

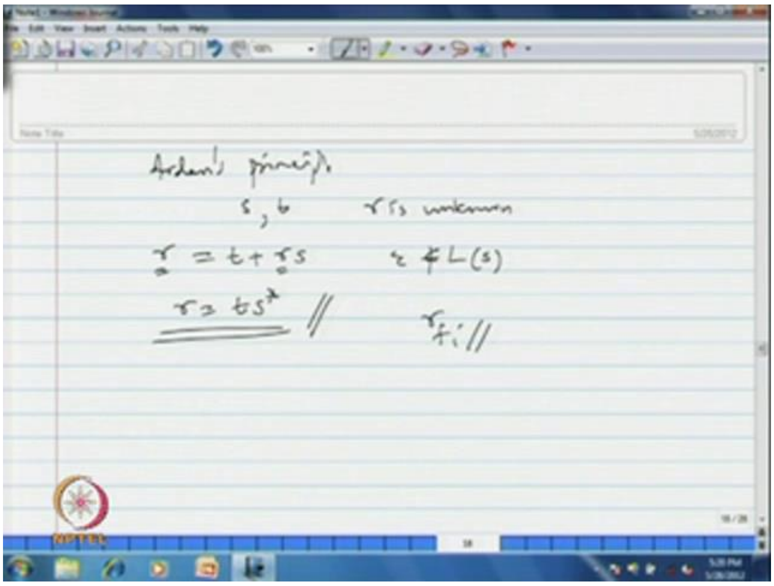
(Refer Slide Time: 47:11)



So, we can write a system of characteristic equation of a as its r 0 is basically r 0 s (0, 0) plus r 1 s (1, 0) plus r i s (i, 0) plus like that r n s (n, 0) plus epsilon. Then, r 1 can be written as r 0 s (0, 1) plus r 1 s (1, 1) r i s (i, 1) plus r n s (n, 1) and so on. For r j you can write it as r 0 s (0, j) plus r 1 s (1, j) r i s (i, j) plus r n s (n, j). Similarly, for r n it is r 0 s (0, n) plus r 1 s (1, n) plus like that r i s (i, n) eventually; r n s (n, n). So, this is how we can write the characteristic equation for each state q 0 through q n we have because 1 of the regular expression r 0, r 1 up to r n.

Now, the system can be solved, this system can be solved for r f i s, because r f 1, r f 2 up to r f k, these are all set of final states. And, since the language of these both D F A is nothing but r f 1 plus r f 2 up to plus r f k solving for these r f i s. We can eventually find out the language regular expression for a language of the D F A a so, this is the corresponding regular expression.

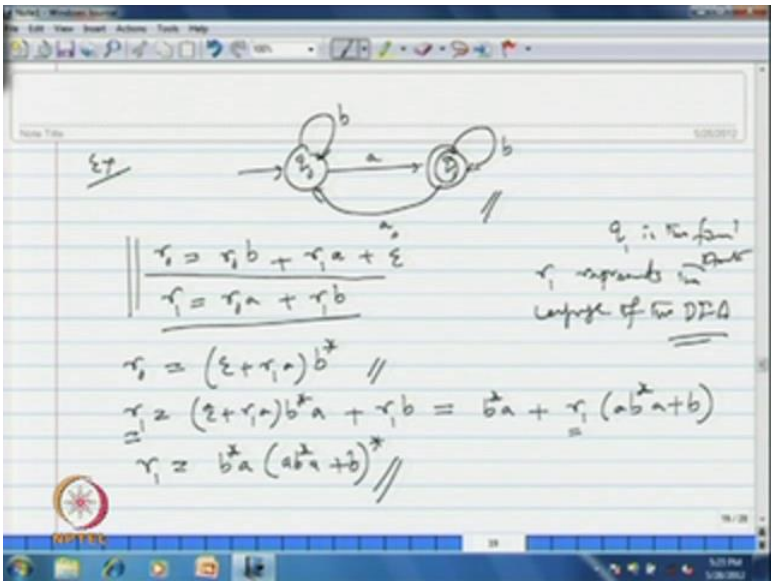
(Refer Slide Time: 50:19)



Now, we can solve it for solve the system equation for r f i s that means; for the final states via state for all the substitution, accept the same unknown may appear in both sides, and both the left and right hand side equation same unknown may appear. So, this situation can be handled using one principle called Arden’s principle. Arden’s principle which says that; if s and t are regular expression and r is an unknown. And, equation of the form r equal to t plus r s where, this unknown appears on both side left hand side and right hand side. And, where epsilon does not belong to language of s so, it has unique solution given by r equal to t s star.

The solution is basically r equal to t s star so, this is what is called Arden’s principle. And, you can use the Arden’s principle whenever; this unknown r is appears in both side of the equation. So, why this while successive suspicions an application Arden’s principle; we can evaluate a expression for final state in terms of symbols from sigma. Since, the expression or the operations involved here are admissible for the regular expression. We eventually obtained regular expression for r f i.

(Refer Slide Time: 51:53)



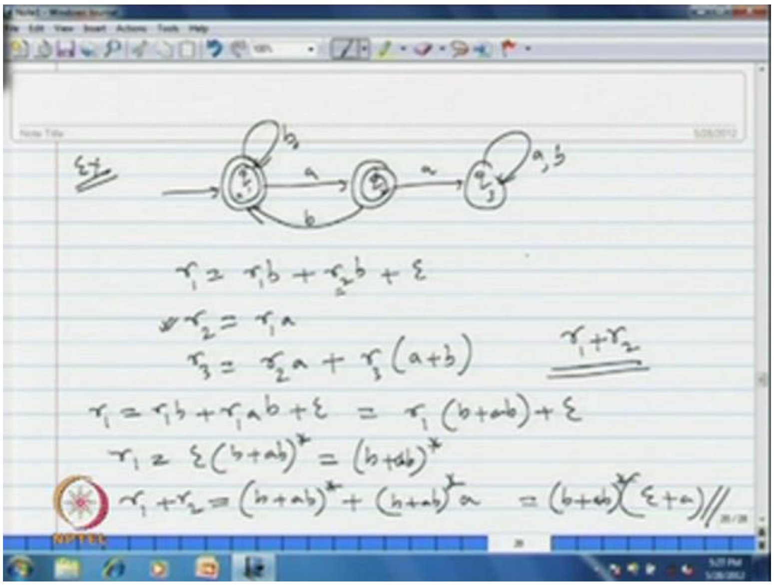
So, we demonstrate this by an example consider the D F A containing only 2 states; where, q 1 is a final state q 1 is a q 0 is a start state. On a it goes to final state, on b it remains the sense that q 1 on remains the same state, and on a it comes to the start state. Now, the characterization for the D F A will be r 0 equal to it is r 0 be it takes r 0, and on b it will even in the same state plus it will be r 1 a. Because from this state on a, we come to this state r 0. So, it is r 1 a plus epsilon, because this is the initial state.

So, this is what we have got a characteristic equation for the start state regular expression for the start state. Similarly, for state q 1 the equation is it is for r 1 is r 0 a plus r 1 b since, q 1 is a final state, r 1 represents the language of the D F A. Because these are only 1 final state that is q 1 is the final state is a only final state. Hence, r 1 represents the language of D F A so; it will solve these 2 equations for r 1. So, we will solve these equations for in terms of for r 1 in terms of a and b, which are the only symbols in the alphabet.

Now, where Arden’s principle we will see that; reconstruct first equation r 0 equal to r 0 b plus r 1 a plus epsilon r 0 can be written as epsilon plus r 1 a b star. Now, substituting this in the second equation here, we see that r 1 is r 0 a means; epsilon plus r 1 a b star a plus r 1 b so, here is r 1 b. Now, simplifying this we can write it as b star a plus r 1 a b star a plus b, this is as by simplifying then by applying again since r 1 appears on the both right hand side and left hand side. Again applying Arden’s principle; on r 1 we get r

1 to be b star a a b star a plus b star. Now, which is a regular, which are these are regular expression represent the language for the given D F A in the example.

(Refer Slide Time: 55:39)



Similarly, if we consider a next example; say we have 3 states q 1 is the start state on a it goes to q 2 which is also a final state, q 1 and q 2 are final states on b it remains the same state q 2. On a goes to q 3, which on a b remains in the same state and, q 2 on b goes to the start state. For this we can write the characteristic equation as; r 1 for the state q 1 as r 1 b, because r 1 on b remains in the same state plus from q 2 we can come to q 1 on b. So, it is r 2 b and from q 2 we cannot come to q 1. And, since this is the start state epsilon will be there then for state q 2 r 2 can be written as from r 1 you can go to q 2 On a. Therefore; it is r 1 a from q 2 you cannot go to q 2, and from q 3 you cannot come to q 2. Therefore, there is only term on right hand side similarly, r 3 can be written as from q 2 I can come to q 3 on a. So, it is r 2 a and from q 3 on a and b both a and b a or b I can come to q 3 again so, r 3 a plus b.

Now, since q 1 and q 2 are final states; the expression for the regular expression r 1 plus r 2 will represent the language for this given D F A. So, we will solve these equations for r 1 and r 2 in terms of a and b now, if we substitute r 2 in r 1 so substitute this r 2 in r 1. We will find that r 1 equal to r 1 b plus this r 2 can be represented by r 1 a b plus epsilon. This is nothing but r 1 b plus a b plus epsilon. Now, applying Arden’s principle you will find that r 1 can be written as epsilon b plus a b star, which is nothing but b plus a b star.

Thus r 1 plus r 2 can be written as b plus a b star plus since r 2 is r 1 a. Therefore; it is b plus a b star a hence, the regular expression for this given D F A is nothing but this is simply b plus a b star epsilon r a. Therefore, this is a corresponding regular expression for the given D F A. Therefore; using Arden's principle and solving the characteristic equation for the given D F A in terms of the symbols of the language, we can always find out the equivalent regular expression for the g D F A, by using this brzozowski algebraic method.