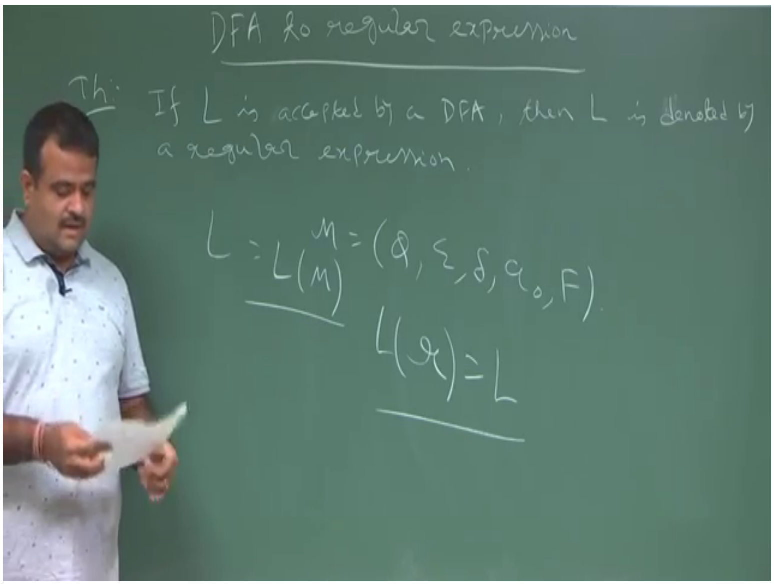
Introduction to Automata, Languages and Computation
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 21
DFA to Regular Expression

(Refer Slide Time: 00:19)



So, we are talking about the Regular Expression and in the last class we have seen the how there is a regular expression is corresponding to a epsilon NFA. So, given a regular expression we can construct epsilon NFA, there is epsilon NFA such that the language accepted by that epsilon in the face that same as the language accepted by the regular expression. That means, regular expression and the epsilon and if you are equivalent.

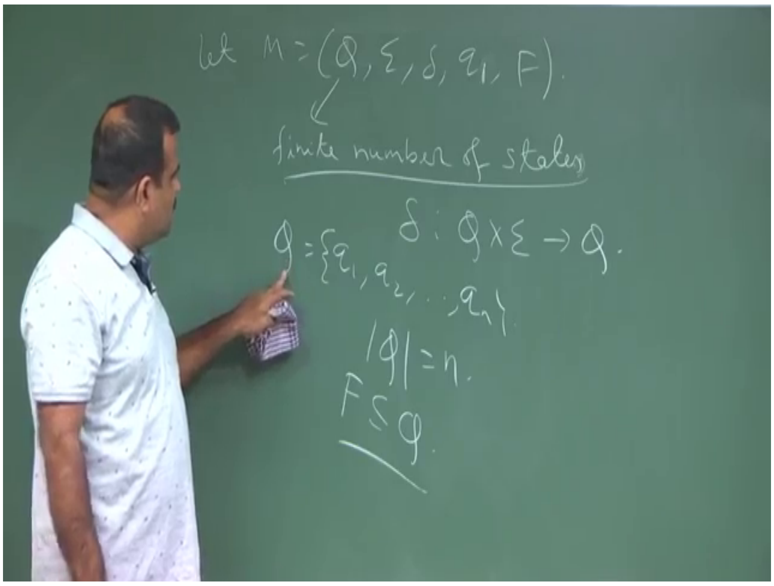
Now, today we talk about the equivalency between the DFA and the regular expression; that means, given a DFA we have a language which is accepting by the DFA. Now, corresponding to that language we can construct a regular expression whose language is the same as the language of the DFA.

So, let us write this in a theorem. So, if L is a L is a regular language which is basically L if L is accepted by a DFA here, then L is denoted by a regular expression ok. So, this is the statement of the theorem. So, given a L which is accepting by a DFA; that means, that is a DF M q 0 such that this L is nothing, but the language of this language of this

DFA, then we can prove that I mean we can construct a regular expression R such that whose language is same as this L.

So; that means, the DFA and the regular expressions are equivalent. So, given the DFA we can construct it regular expression which is accepting the same language. The language you know language corresponding to the regular expression is same as the language of the automata ok. So, we have to prove this. So, to prove this we take a DFA and DFA we know it is a finite state machine. So, the state space is DFA is finite state space of DFA finite ok.

(Refer Slide Time: 03:19)

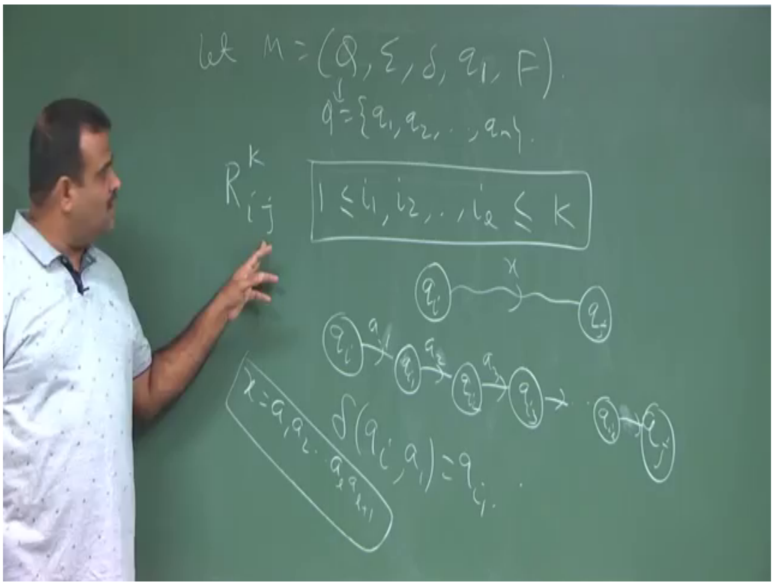


So, let M is a DFA which is having this couple q 0 or this we can start with q 1 because I will explain and F. So, this is a Q is a finite number of states and sigma is a finite number of alphabet and this is a delta is the rule we know delta is nothing, but the transition rule and q 1 is the starting state. We take q 1 because we want to. So, because this the state is finite. So, without any loss of generality we can assume the states are q 1 q 2 q n if the cardinality of q is n. Suppose there are n state, n could be anything, then we can rename the states. There is no nothing there is there is no harm of renaming the state.

So, that state we are renaming at starting from q 1 to q n. We could do it from q 0, but anyway the it will help us to prove this theorem. So, our states are numbered by q 1 to q n. Even we can do it 1 to n, states are 1 2 n; 1 means q 1, 2 means q 2 like that ok. So, states are numbered by 1 to n q 1 to q 1 and q 1 is we are taking as a starting state and F is a subset of this subset of this. So, F is also some consists of some of this q i's which is the accepting state or the final state. So, this we can assume, I mean by renaming the states we can take this by this numbering ok.

Now, this is our state space; this is our state space this q and now we define some notation which is denoting by R i j k. So, now, Q is that is right.

(Refer Slide Time: 05:55)

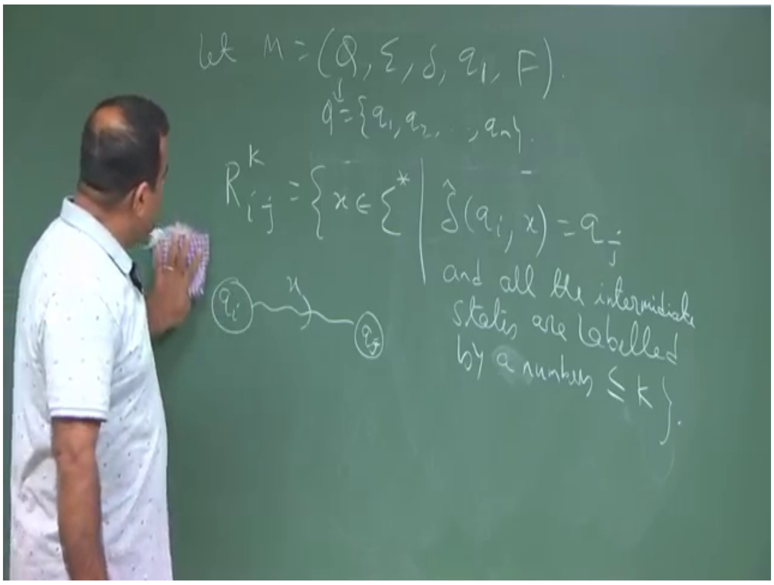


This Q is q 1 q 2 q n ok, you want into given ok. Now, we denote this R i j k; this is basically the set of all string we are at q i set of all string which is starting from q i and which is ending to q j; this is q i into q j and what is this k means? So, that means, it will visit some state by this rule delta. So, this is say q q i 1 q i 2 this is a 1 into something like that or a i 1 in general a 1 a 2; a 1 a 2 like this so, q i 3 a 3 dot.

Some number say q i l which is basically our this then we go to q 0. So, these are the intermediate states direct arc; these are the intermediate states coming from the rule delta. So, that means, delta of q i comma a 1 is q i 1 like this delta of q i 1 comma a 2 is q i 2 like this. So, this way we go. So, this is the and now what is this k means; k means the number, this is i 1 i 2 these are the states we are visiting in the intermediate path. So, this number has to be less than k. So, that is the restriction. So, this number has to be less than k or less than equal to k.

So, this all these i 1 i 2 i l has to be less than equal to k. We know they are greater than. So, this is the restriction. So, these constraints of set of all string x; so, x is nothing but a 1 a 2 a a l then a l plus 1. So, this is set of all string which is starting with the state q i; this is i and at the end of the string it is reaching to the state q j and the intermediate these are all intermediate state and the intermediate state the number which we are seen is strictly less than equal is less than equal to k. So, that; that means, intermediate we are not seeing any state which is more than k that is the restriction that is the meaning of this and there is no restriction on i and j with k, i j could be greater than k; all the thing the intermediate node we have a restriction, there is no restriction of i n j ok.

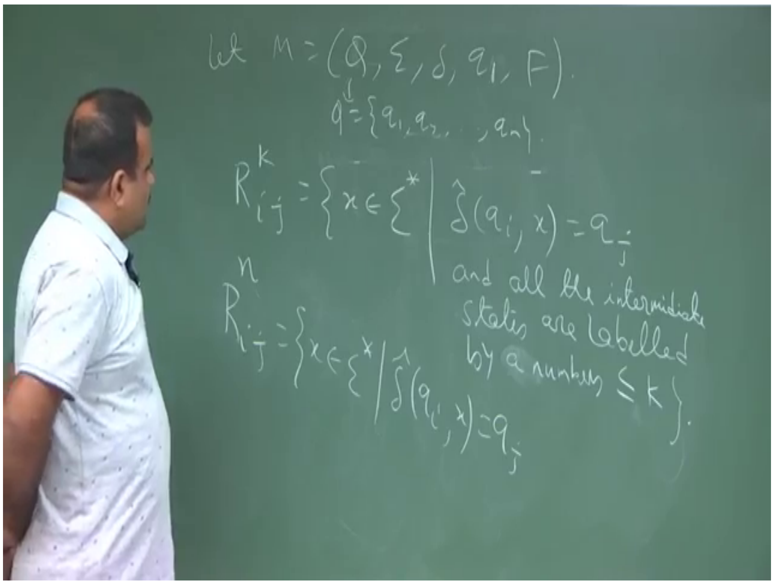
(Refer Slide Time: 09:37)



So, let us write this in a in a form like this. So, this is the set of all strings such that delta hat because this is a string delta hat of q i comma x is equal to q j and. So, we are starting from q i and with this x we are reaching to q j at the end of the string and all the intermediates nodes has to be labeled less than equal to k; all the intermediate states are nodes; states are labeled by in the by a number less than equal to by a number by numbers less than equal to this is the symbol R i j k.

So, we are starting with i'th state q i we are reaching to j'th state, but in the intermediate state we are not visiting any node which is more than number more than k. So, that is the this.

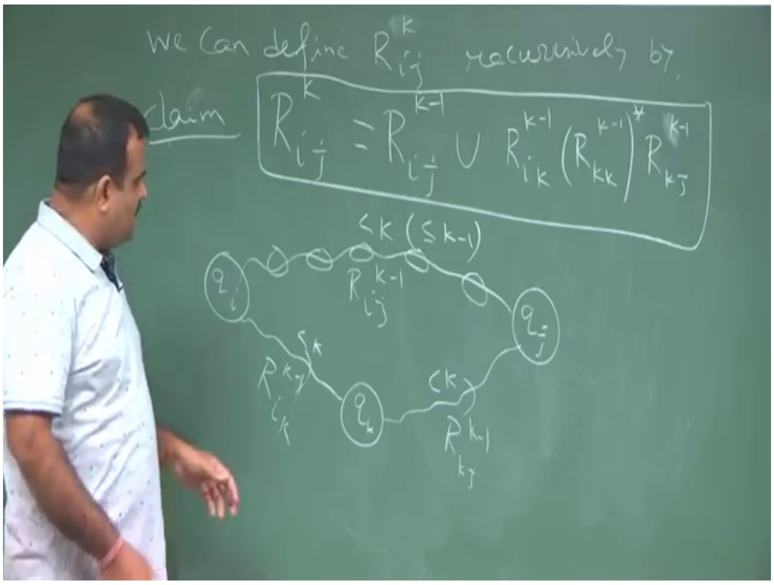
(Refer Slide Time: 11:13)



Now, if k we put n, so, what do you mean by R i j n? R i j n means we are we are starting with this is the set of all string we are starting with q i we take will go to q j and the intermediate not we are not seeing any node which is more than label more than n, but there is no node which is level more than n. So, this is the set of set of all string. There is no restriction, set of all string we start with q i which can reach us to q j that is all. So, that is the meaning of this ok.

So, if we put n then it will come this part will not come because n is the all state all the states are less than equal to n ok. So, this is our notation R i j k. Now we will have a we will try to get a recurrence on this notation recurrence relations and then we will have a regular expression for this language. So, let us just write the so, we have we have a recursive relationship on R i j.

(Refer Slide Time: 12:41)



So, this we can define. So, you have to prove this is our claim. We can define R i j k recursively by this relation. R i j k is equal to R i j k minus 1 this is 1 case union of R i k k minus 1 R k k k minus 1 this is star and R k j k k minus 1 ok; this is the this is our this is the recursive relation for this R i j k; we to prove this is our claim this is our claim. Let you prove this ok.

So, yeah so; that means, what? This means what? This means we have starting from q i this is the set of all strings starting from q i we have to go to q j. Now there are 2 possibilities. One is and in the intermediate path we do not want to see any node which is labeled greater than k, we are only allowed those nodes those states whose label are greater than n less than equal to k. So, there are 2 possibilities; one is there will be no node of label by k.

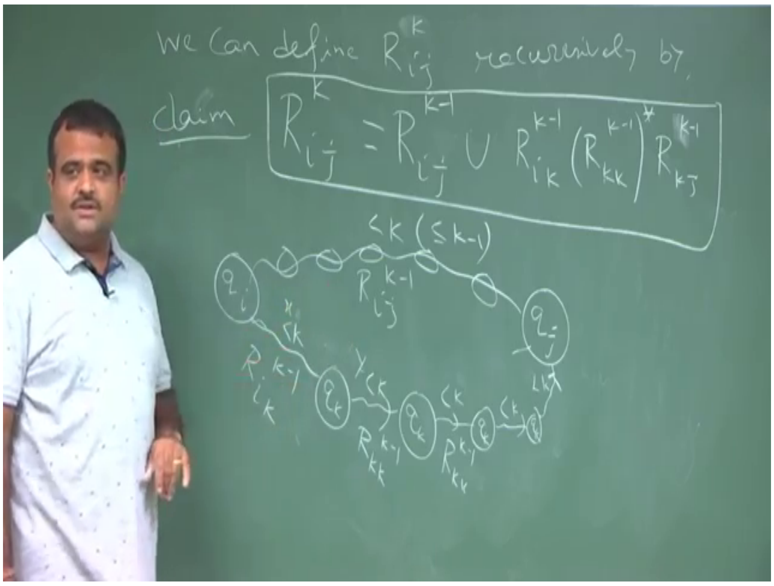
So, one path is like this. All the nodes will be less than k all the notes we are visiting over here how less than k. So, we have to take although such string which is starting from here going to here and which is not seeing any node which is equal to k. So, that means, we are seeing only the nodes which are less then strictly less than k. So, that collection of the string denoted by this ok; less than k means less than k means less than equal to k minus 1. So, this is the set of all string R i j k minus 1 is we are starting from q i we are going to q j with those strings which are not saying any note which is k. So, that is this way one this is one possibilities and another possibilities is we have a node which is labeled by k. So, q k is there.

So, if q k is there then we can go to q k and from this q k we can go here and there are few possibilities here also sub possibilities that there may be only one q k in this path or there may be many q k's in this path. So, if there is only one q k in this path then there is no k k no q k here. So, this will be denoted by all the nodes are strictly greater than k, all the nodes are strictly sorry strictly less than k strictly less than k.

So, this will be denoted by R i k k minus 1 and this will be denoted by R i R k j k minus 1. So, concatenation of these 2 if there is only 1 q k in this path. So, this will be denoted by this without appearance. This the star means it may come may not come ok, in that case this is this from concatenate with this.

Now, if there is more q case in that path then what will happen? If there is more q k's in this path, then we can just have k's like this.

(Refer Slide Time: 17:17)

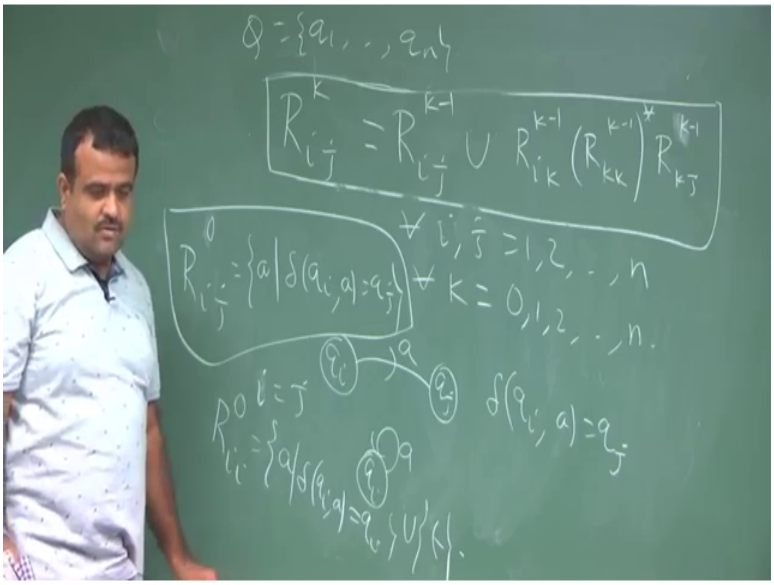


So, suppose this is the first q k then this is the second q k like this there may be 2 or more q k like this q k like this then we have. So, there is no q k over here. So, these are all less than k there is no q k over here less than k less than k less than k ok. Now this is the path R i k k minus 1 and this is the path R this is the path R k k k minus 1 because there is no q k in this path. This is also R k k k minus 1, but this is a loop. So, that is why we have put in a star. So, it may.

So, this is our x 1 this is x 2 and this sorry this is our x this is our y and this y may repeat, y star; y concatenated with y concatenated with y some another y, but that y is coming from this because there is no q k in this path there is no q k in this path like this. So, that will eventually give a this star. Star means it star staring both 0's also. So, the 0's means no appearance that is the case where only 1 q k present. If there is 2 q k then we have this 1 if there is 3 q k we have 2 3 4 like this. So, this is the star concatenation of this. So, this is the informal proof of this. This is if there are 2 ways: One is the string like all the path is having no node which is usable by k we have a path with node label by k, but that may come 1 time 2 time 3 times. So, that is why we are putting this. So, this is the proof of this ok.

So, we take this and this is true for all i and j. So, this result is true for this result is true for all i j comma 1 to n and all k.

(Refer Slide Time: 19:51)

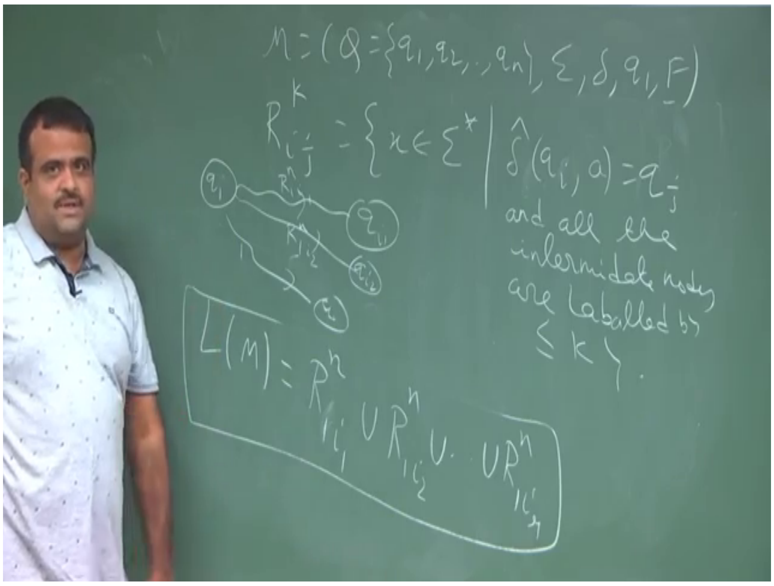


So, k can be 0 here for all k. So, what is the meaning of k 0? What is R i j 0? R i j 0 means set of all strings. So, here at q i we have to go to q j and the intermediate node the intermediate node we will see must be label less than equal to 0, but our states are all level by q 1 to q n. So, there is no states which is q 0; so, that that means, there is no intermediate node in this path. So, ; that means, the direct path will come direct arc will come.

So, if we have a arc from q i to q j direct edge. So, that means, delta F naught delta hat delta of q i a is equal to q j yeah. So, then this is this is this consists of all such a i’s; all such a i’s such that delta of q i a from equal to q j; this is our R i j if R i j 0. If i not equal to j, but if i is equal to j if i is equal to j then what will happen? That means, R i i 0; so, we are at q i you want to go to keep there is self loop a then otherwise it is epsilon. So, it consists of all such a such that delta of q i a q i union of epsilon this is our R i i ok.

Now, we will take this result to have a regular expression corresponding to this language. So, that we are going to show now, ok. So, just to recap what we have.

(Refer Slide Time: 22:55)



So, we have given a given a DFA we have given a DFA, we can construct a R i j which is nothing but set of all strings such that delta of delta hat q i comma a is nothing but q j and all the intermediate nodes labeled by less than equal to k ok. Then we have seen this recurrence and then what is the then form for [FL]; suppose F consists of these are the state; q i 1 q i 2 q i R, suppose they are the final state of the automata then by this notation what is our what is our [FL], what is our language? So, then the language is nothing, but. So, we start with 1 that is a and we go to any stream which can start with 1 and which can reach to one of the final state this is 1 final state and we have no restriction on the intermediate nodes. So, that will give us the set of all possible possibilities which can start with q 1 and which is going to i 1 union of R 1 i 2 n union of R 1 i R n. So, this is our language. This is the language this is the language if this is all the accepting states accepting states then we start from the starting state with this q 1.

So, we start from q 1 q 1 and any stream which is reaching to 1 of this final state q of i 1. So, this is R 1 i 1 n because we do not want to put any restriction on the states we want to visit all the states. So, this is q of i 2 like this. So, this is R of 1 i 2 n like this we continue. So, this is q of i 3 dot dot q i r. So, this is the language now we are going to in the next class we will show that this will corresponding to a regular expression. So, we have a regular expression which will which will accept this I mean whose language will be corresponding to this. So, that will discuss in the next class.

Thank you.