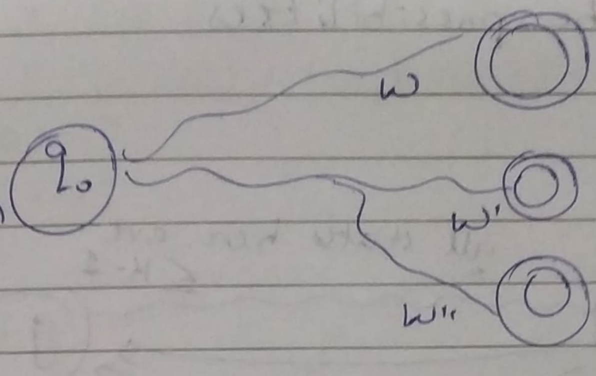
We will show that for every DFA, there exists a regular expression such that it accepts the language of DFA

DFA ≤ reg. expression

A DFA has a start states and a set of end state

L is language of this DFA

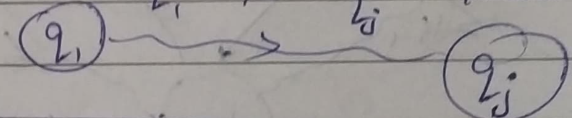


$w \in L$

$w' \in L$

$w'' \in L$

For a q_i to q_j transition, let input be a, b, c, ...



so $\boxed{abc\ldots = x}$ x is label of that path

Let us also label states as $1, 2, 3, \ldots, n$

R_{ij}^{k} is the set of all strings that take us from q_i to q_j without going through any state labelled $> k$.

$$L(M) = \bigcup_{z \in f} R_{1z}^{n}$$

final states

R^n is meaningless here as $n > k$ is not present.

set R_{1z}^{n} is represent by reg. exp. as r_{ij}^{z}

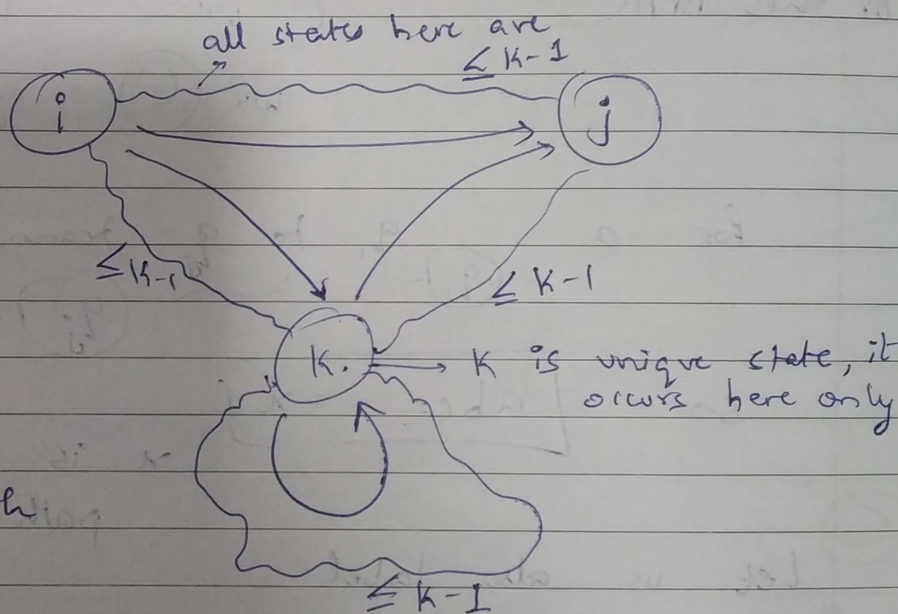
$$L(M) = \bigcup_{z \in F} r_{1z}^{n}$$

Understanding more on R_{ij}^{k}

we cannot use states $> k$

but can ~~touch~~ touch $k, k-1, k-2, \ldots, 1$.

What can be the possibilities.



path having intermediate states.

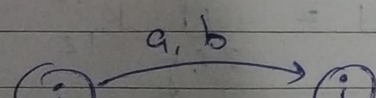
direct path

$$r_{ij}^{k} = r_{ik}^{k-1} \left(r_{kk}^{k-1}\right)^{*} r_{kj}^{k-1} \cup r_{ij}^{k-1}$$

we can recurse and lower the $k, k-1, \ldots$

boundary case r_{ij}^{0}

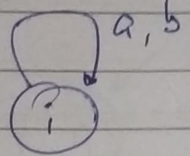
$\cancel{r_0}$ r_{ij}^{0}



$$R_{ij}^{0} = \{a, b\}$$

$$\delta(i, a) = j$$
$$\delta(i, b) = j$$

another boundary case R^0_{ii}



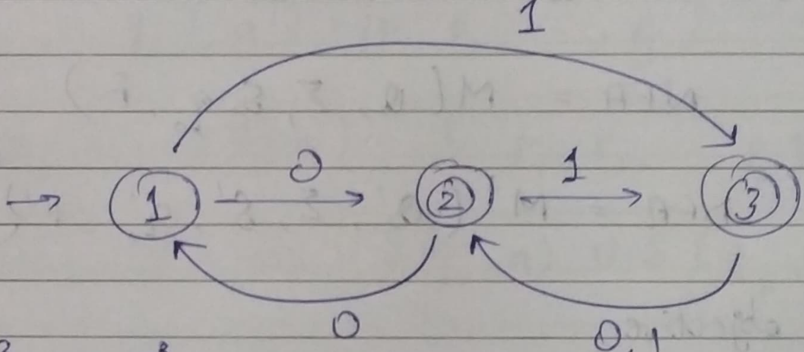
$R^0_{ii} = \{ \emptyset, a, b, \varepsilon \}$

it means we stay there. no transistion

$r^0_{ii} = a \cup b \cup \varepsilon$

$\delta(i, a) = i$
$\delta(i, b) = i$

eg. DFA



$L(M) = R^3_{12} \cup R^3_{13}$

$L(M) = r^3_{12} \cup r^3_{13}$

$r^3_{12} = r^2_{13} (r^2_{33})^* (r^2_{32}) \cup r^2_{12}$

$r^2_{13} = r^1_{12} . (r^1_{22})^* (r^1_{23}) \cup r^1_{13}$

$r^1_{12} = r^0_{11} (r^0_{11})^* (r^0_{12}) \cup r^0_{12}$

$= (\varepsilon (\varepsilon)^+ 0) \cup 0 = 0 \cup 0 = 0$

Hence we have proved for every DFA we have a regular expression.

- To prove: NFA $\leq$ DFA

We will prove

NFA with ϵ transistions --- NFA with no ϵ transistions --- DFA

First we will prove this

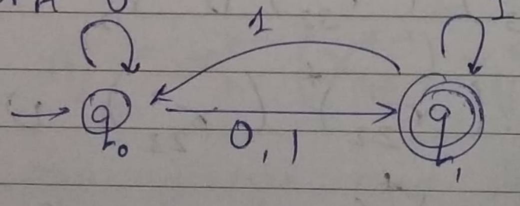
- Assume no ϵ transistions.

$$NFA = M(Q, \Sigma, \delta, q_0, F)$$ corresponding language $= L(M)$

$$DFA = M'(Q', \Sigma, \delta', q_0', F')$$ → language $L(M')$

objective
$L(M) = L(M')$

- Observing by an example.

NFA


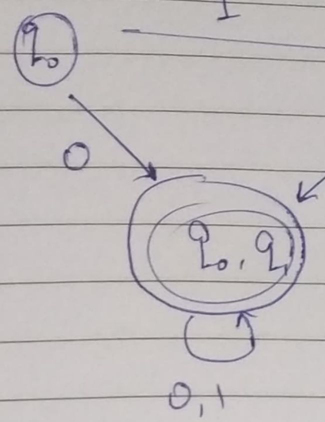
$\delta(q_0, 0) = \{q_0, q_1\}$
$\delta(q_0, 1) = q_1$
$\delta(q_1, 0) = \phi$
$\delta(q_1, 1) = \{q_0, q_1\}$

DFA
we will consider each combination in power set of states as a distinct state.

$(q_0), (q_1), (q_0, q_1)$ These 3 are distinct states.

ϕ is also a possibility.

Few, transistion are clearly visible from NFA, but for others we make it here



$\delta'(\{q_0, q_1\}, 0) = \{q_0, q_1\} \cup \phi$

directly expended here.

$= \{q_0, q_1\}$

see here for method

$\delta'(\{q_0, q_1\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1)$

$= \{q_1\} \cup \{q_0, q_1\}$

$= \{q_0, q_1\}$

$\delta'([q_1, q_2, \dots q_i], a) = [p_1, p_2, \dots, p_j]$ → in NFA

$\delta'(\{q_1, q_2, \dots, q_i\}, a) = \{p_1, p_2, \dots p_j\}$ → in DFA

$\delta'(\{q_1, q_2, \dots, q_i\}, a) = \delta(q_1, a) \cup \delta(q_2, a) \cup \dots \cup \delta(q_i, a)$

Comparing defining variables for NFA, DFA

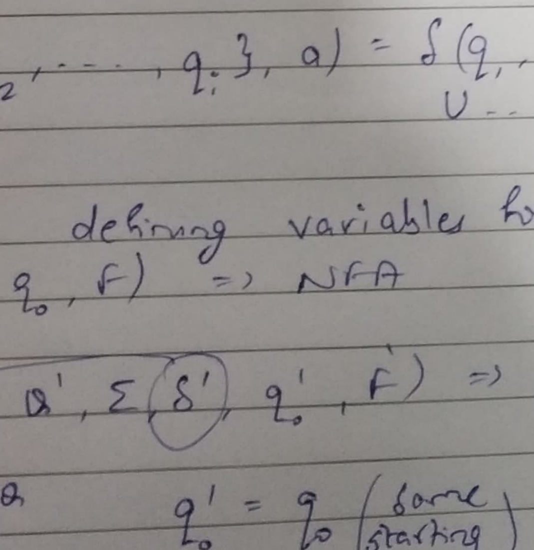
$M(Q, \Sigma, \delta, q_0, F) \Rightarrow$ NFA

$M'(Q', \Sigma, \delta', q_0', F') \Rightarrow$ DFA

$Q' = 2^Q$ $\quad q_0' = q_0$ (same starting point)

$F' \Rightarrow$ any state in Q' that has an element in F.
eg. if q_1 was final state. $\{q_0, q_1\}$, $\{q_1\}$ will be final state in corresponding DFA

argument for final state



By w, there is a way that we can reach a final state.
so $\{q_1, q_2, q_3\}$ will be a final state in DFA.