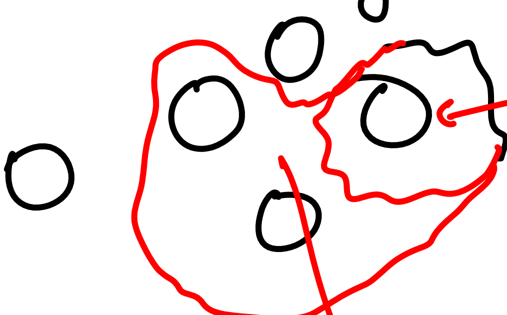
COL 352 Lecture 17 Feb 17

Minimal State DFA

<u>Thm</u> The minimum state DFA for a regular language L is unique upto isomorphism (renaming of states).

This is a corollary of the Myhill Nerode thm and corresponds to the equ classes of R_L

Given any DFA M for L



is completely contained with same equ class of M' (the machine from R_L)

not possible

For any machine M,   So it min state
#states of M' $\leq$ #states of M

Consider a machine M which the same no of states as M'

We want to map the states of M to M'

Let q be a state of M then clearly for some $x \in \Sigma^*$

$\hat{\delta}(q_0, x) = q$ (otherwise M is not min state)

$q \longrightarrow [x]$ (corresponding to R_L)

for consistency verify that for any other y s.t. $\hat{\delta}(q_0, y) = q$

$[y] = [x]$

Since x and y belong to the same state they are in the same eqv class of R_M $\Rightarrow$ they are in same eqv class of R_L

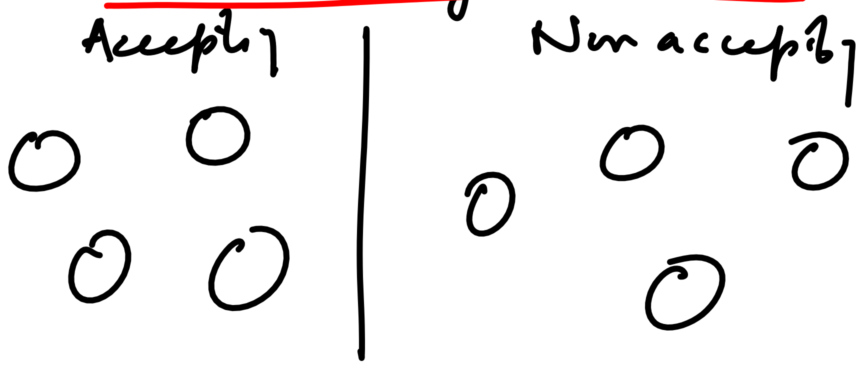
To complete the proof show that

$\delta(q, a) = \delta([x], a) = [xa]$

(this is how M' was constructed

Constructing min state DFA

Since this is related to R_L which is above string x, y behaving identically with respect to acceptance / non-acceptance for any concatenate string z, no accepting state can be equ to non-accepting state because it must satisfy R_L for $z = \varepsilon$

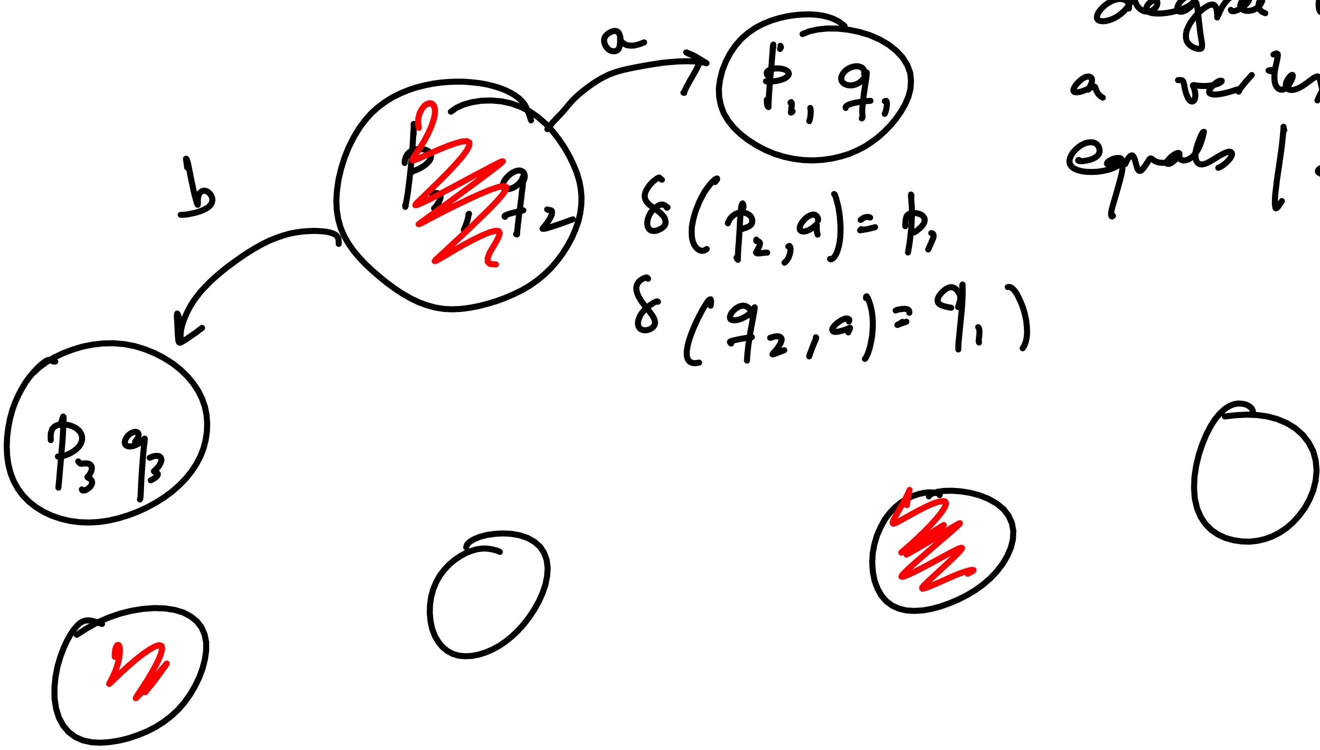


Consider a pair of state p, q
We cannot club p and q if for some string $z \in \Sigma^*$
$$\hat{\delta}(p, z) \in Q - F \text{ and } \hat{\delta}(q, z) \in F$$
or vice versa

$\Rightarrow$ If (p', q') are not equ then p, q are not equ if $\delta(p, a) = p'$
$\delta(q, a) = q'$

Define a graph whose vertices correspond to $Q \times Q$

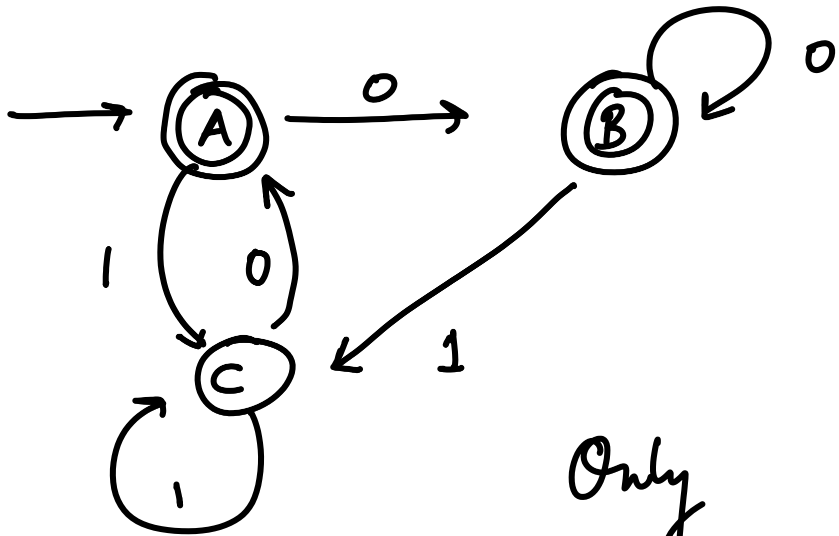
and edges correspond to



degree of a vertex equals $|\Sigma| = k$

Initially 'mark' all pairs known to be not equivalent, viz $F \times (Q - F)$

A special case of "coarsest partitioning problem"



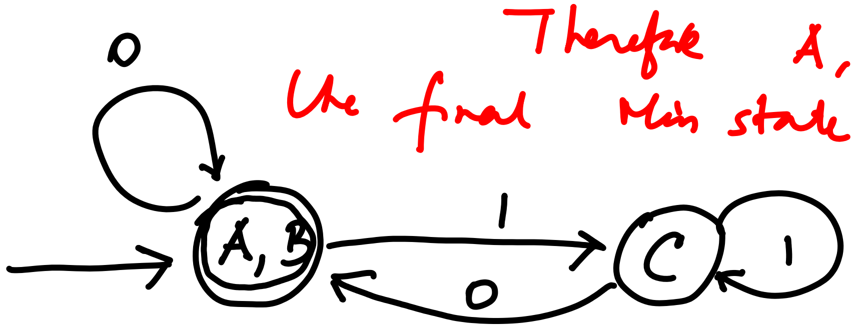
Try to m-this

Only (A,B) can be clubbed

$$(A,B) \xrightarrow{0} (B,B)$$
$$(A,B) \xrightarrow{1} (C,C)$$

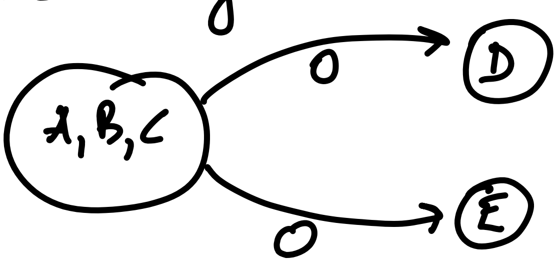
Since the states are same in both cases, they cannot be distinguished by any string.

Therefore A, B can be clubbed & the final Min state automaton will be



The transitions of the new machine can be thought of as graph-contraction, i.e. once vertices are merged, the edges are between merged vertices

Argue why we cannot have a situation



after merging vertices, i.e. no non-determin transitions