DFA minimization algorithm

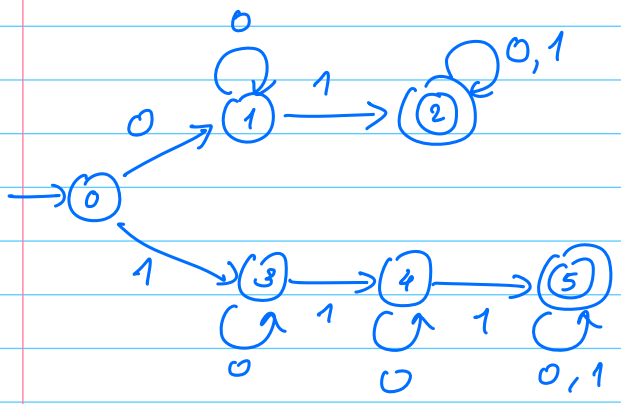
- Initially all pairs of states are unmarked
- Mark $\{q, q'\}$ if $q \in F$ & $q' \notin F$ or vice-versa
- Repeat until there are no changes

  if $\exists \{q, q'\}$ - unmarked s.t

  $\{\delta(q, \sigma), \delta(q', \sigma)\}$ - marked for some $\sigma \in \Sigma$

  then mark $\{q, q'$

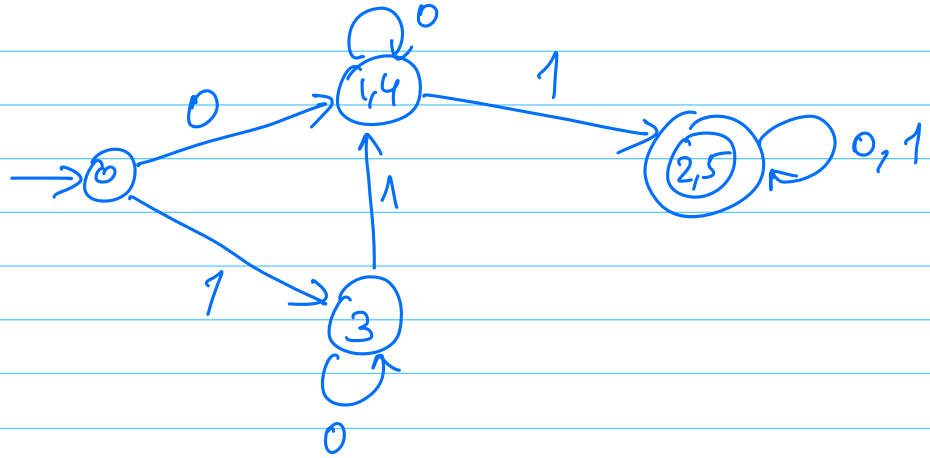
- If $\{p, q\}$ are unmarked iff $p \approx q$



	0	1	2	3	4	5
0		X	X	X	X	X
1			X	X		X
2				X	X	
3					X	X
4						X
5						

$\delta(0,1) = 3$
$\delta(1,1) = 2$

$\delta(0,1) = 3$
$\delta(3,1) = 4$



Regular expressions

- Concisely express patterns
- used for finding strings matching a fixed pattern - grep, lexical analysis..

Inductive definition:

- $\sigma \in \Sigma$ is a r.e
- ε is a r.e
- ϕ is a r.e
- if R_1, R_2 are r.e, then
  - $R_1 + R_2$ is a r.e
  - $R_1 \cdot R_2$ is a r.e
  - R_1^* is a r.e
- What about $\overline{R_1}$?

Given a r.e R, L(R) is the set of strings that match the expression R.

$L(R_1 + R_2) = L(R_1) \cup L(R_2)$

$L(R_1 \cdot R_2) = L(R_1) \cdot L(R_2)$

$L(R^*) = L(R)^*$