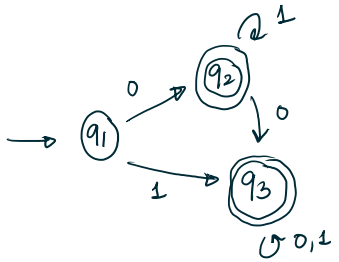
MATH 2301 — 09 October 2020

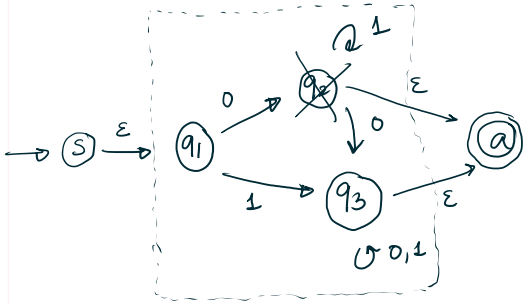
* We have (in principle) converted any regex into an NFA
* Now we try to convert a DFA or NFA back to a regex.

E.g. (last time):



Let's be systematic:

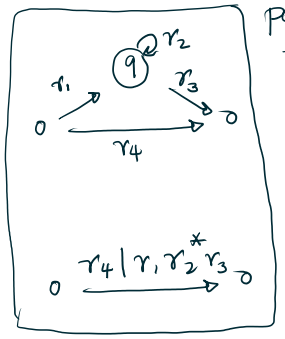
① Simplify by adding a dummy start & accept state:

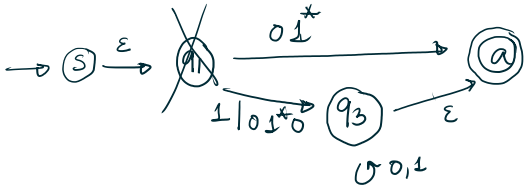


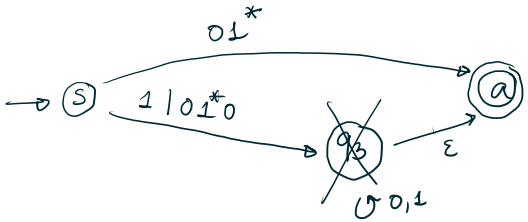
② Delete internal states one-by-one, updating arrows & labels as necessary [in any order]

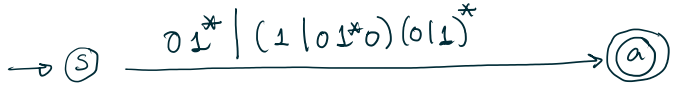
Procedure: delete a state q.

- Look at all possible 2-step paths through q







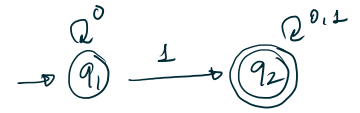


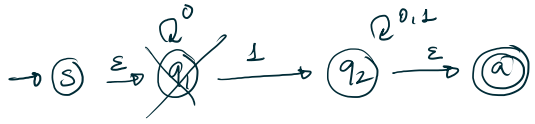
Now we're done, because all internal vertices are gone.

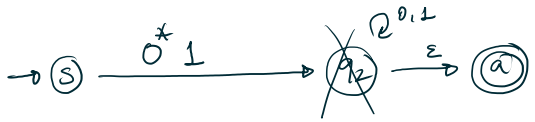
⇒ A regular expression that's equivalent to the original DFA is

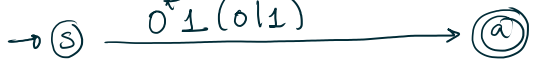
$$01^* | (1|01^*0)(01)^*$$

More examples:

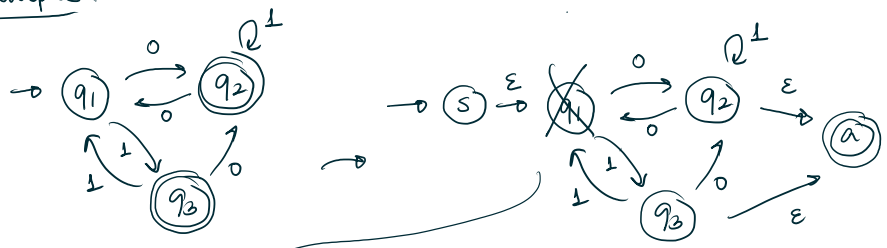


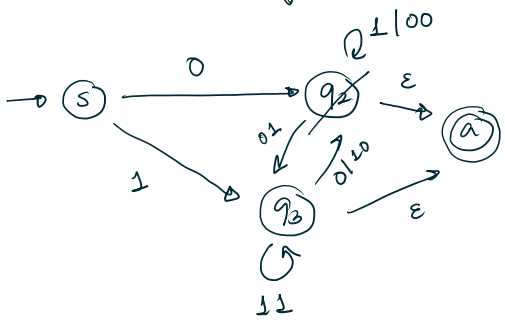






Example:





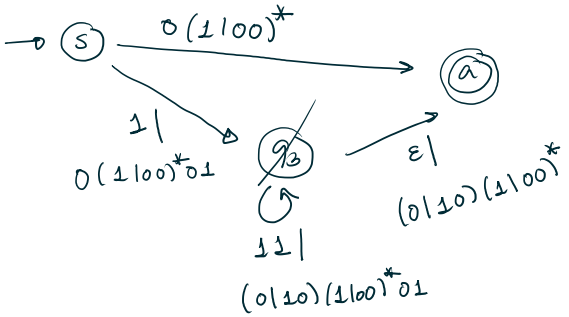
Incoming edges:

- $s \xrightarrow{\varepsilon} q_1$ ✓
- $q_2 \xrightarrow{0} q_1$ ✓
- $q_3 \xrightarrow{1} q_1$

Outgoing:

- $q_1 \xrightarrow{0} q_2$
- $q_1 \xrightarrow{1} q_3$

[6 combinations]

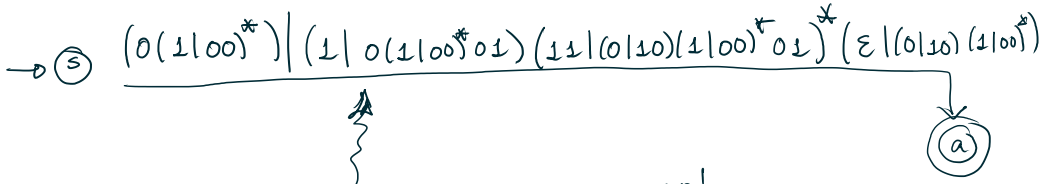


incoming @ q_2:

- $s \xrightarrow{0} q_2$ ✓
- $q_3 \xrightarrow{0|10} q_2$

outgoing:

- $q_2 \xrightarrow{\varepsilon} a$
- $q_2 \xrightarrow{01} q_3$



That's the final answer!

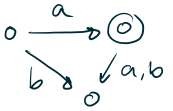
Examples for converting regular expressions into NFAs

Let $\Sigma = \{a, b\}$

$r = (a|b)^* aba$

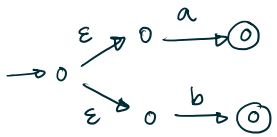
NFA for $\{a\}$:

or



NFA for $\{b\}$:

NFA for $a|b$



(or product construction of DFAs)

NFA for $(a|b)^*$: (Finish this next time)