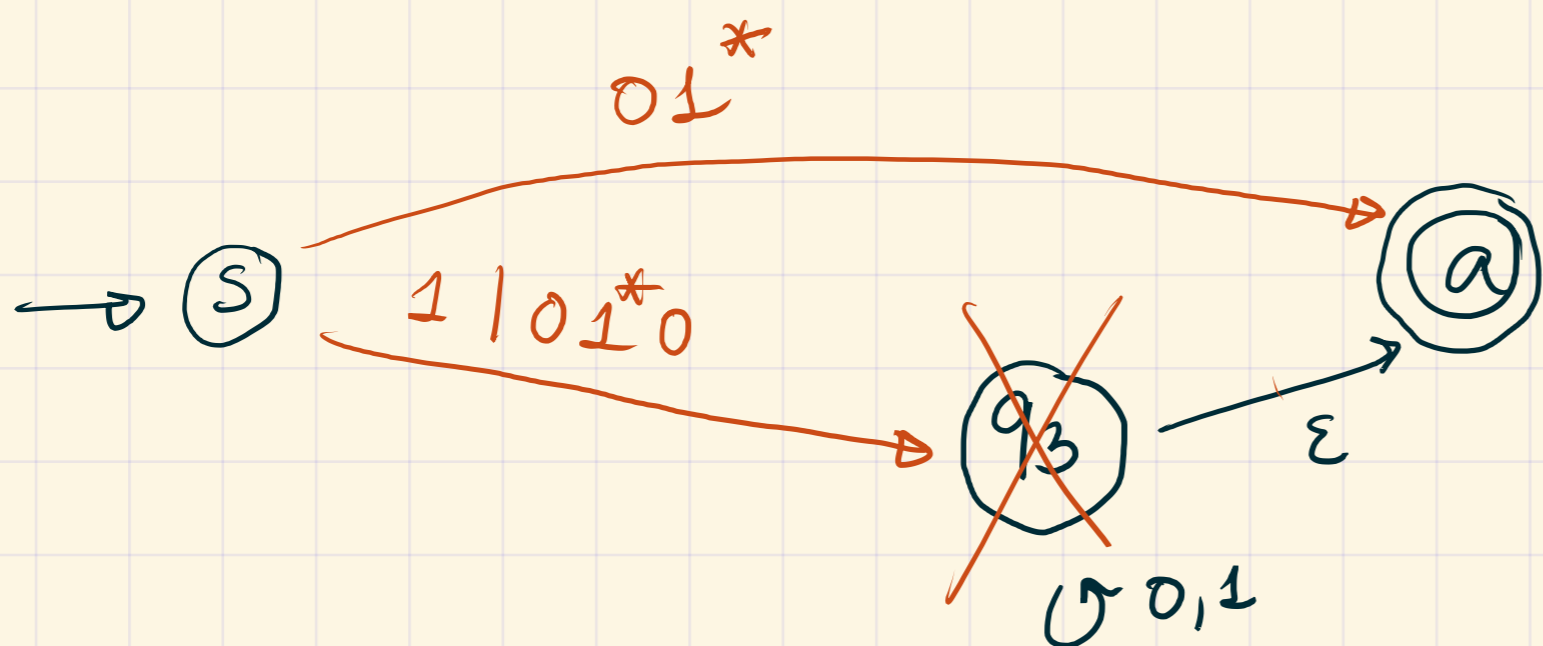
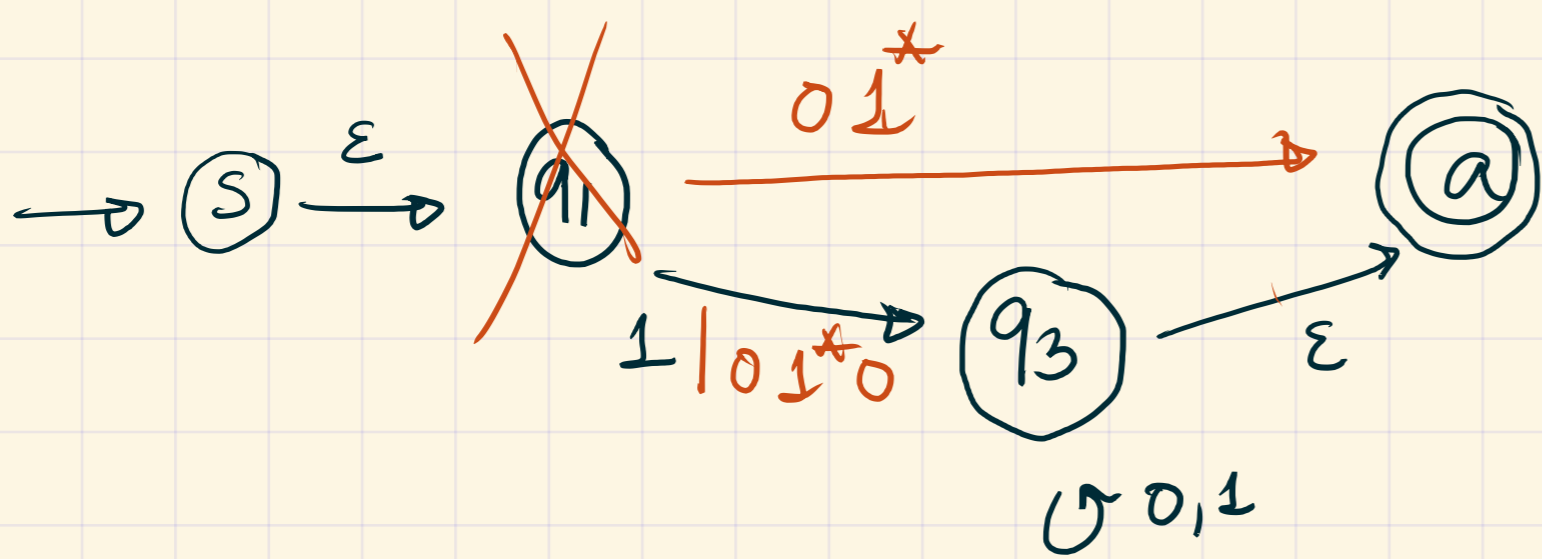
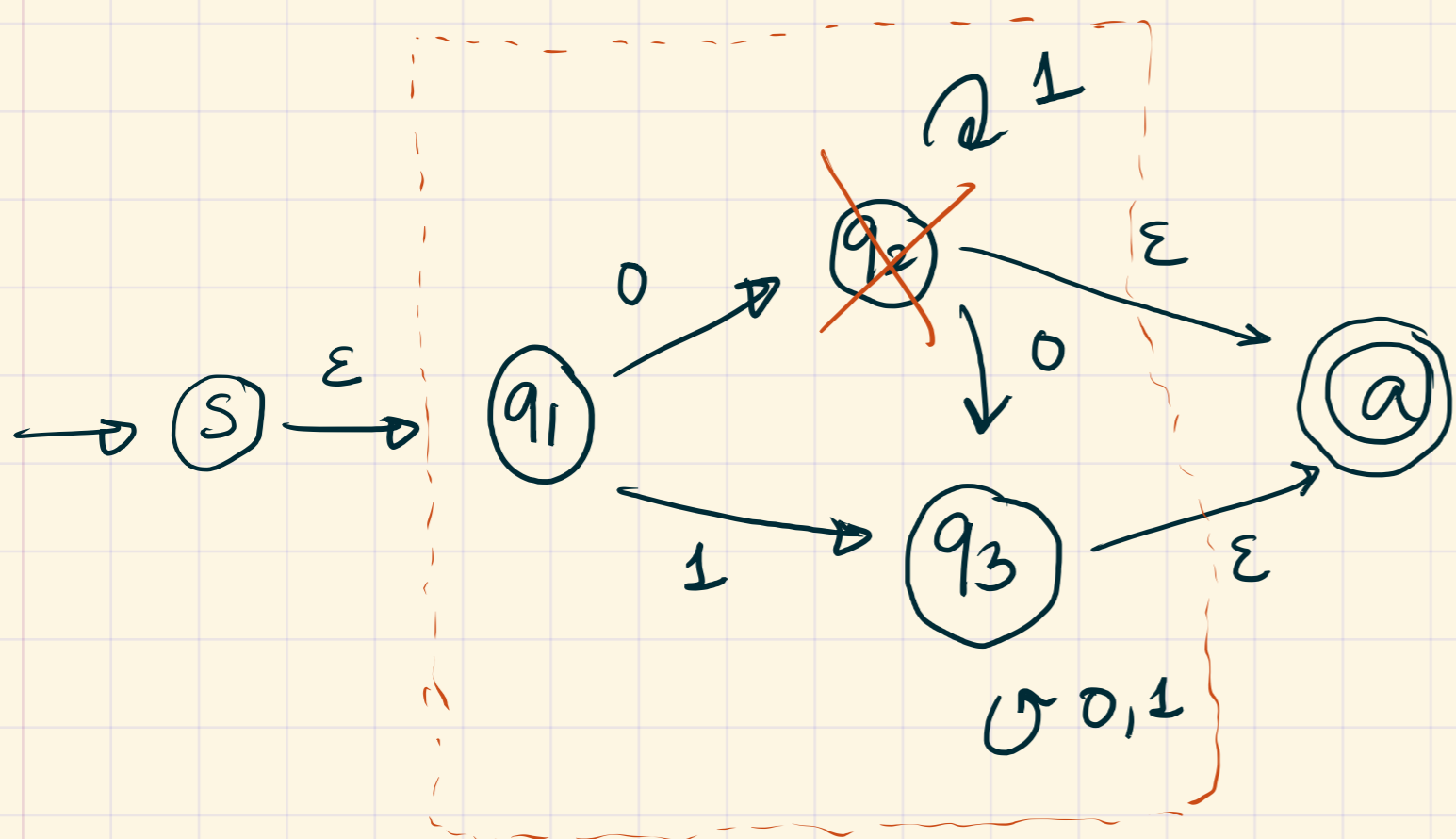
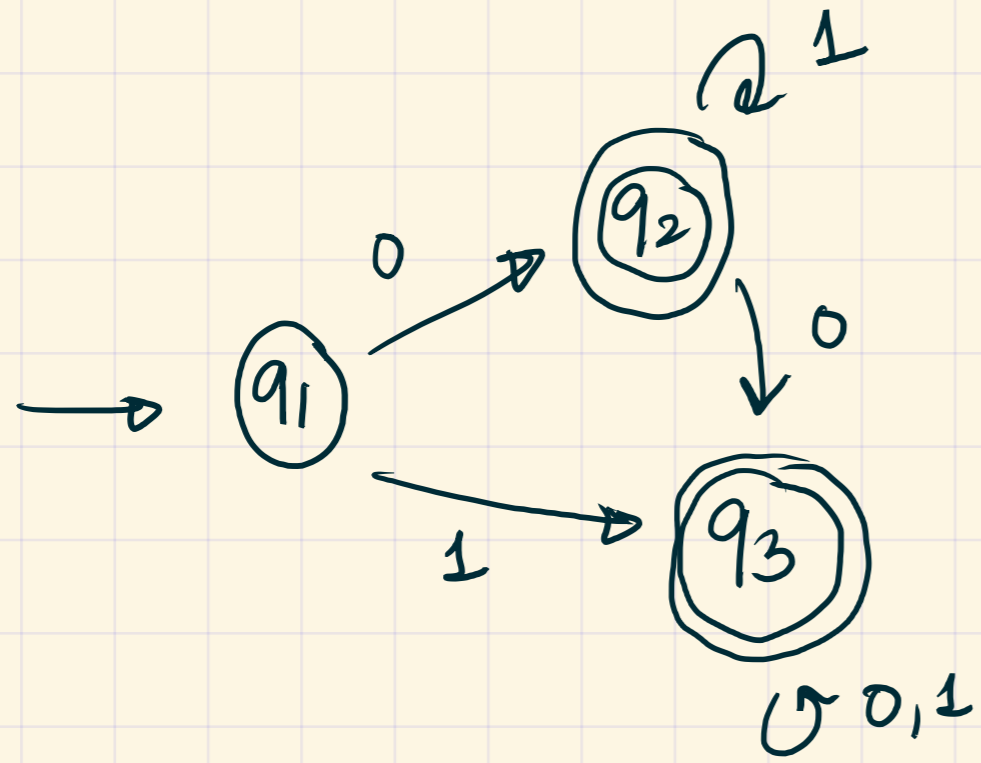


* We have (in principle) converted any regex into an NFA
 * Now we try to convert a DFA or NFA back to a regex.

E.g. (last time):



Let's be systematic:

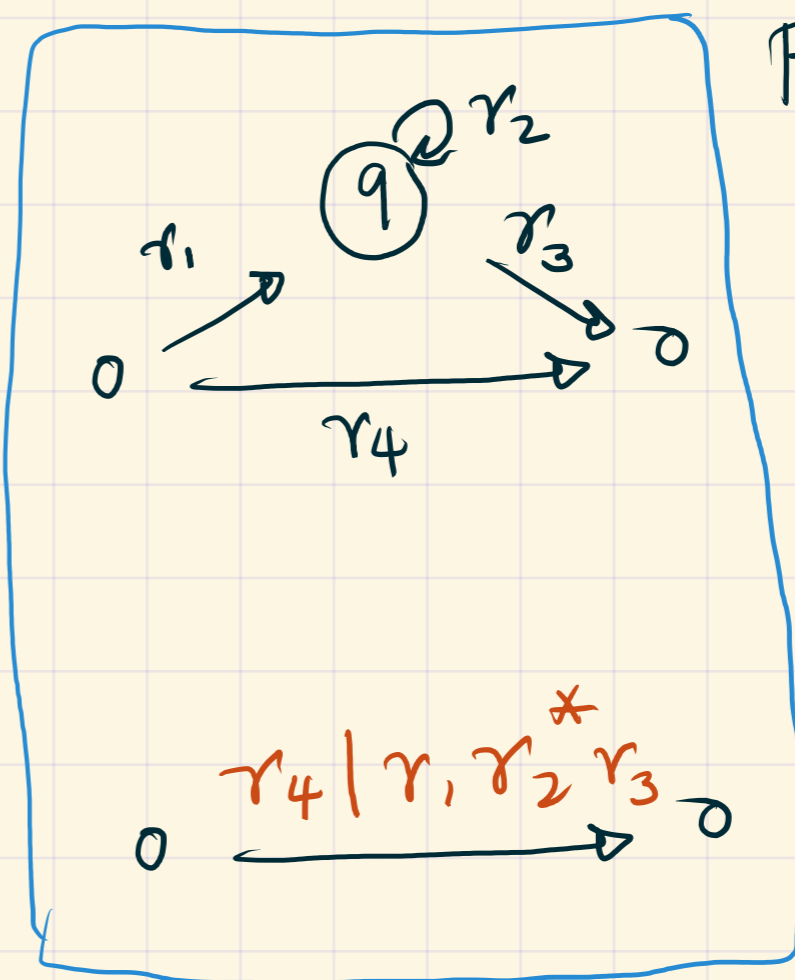
① Simplify by adding a dummy start & accept state:

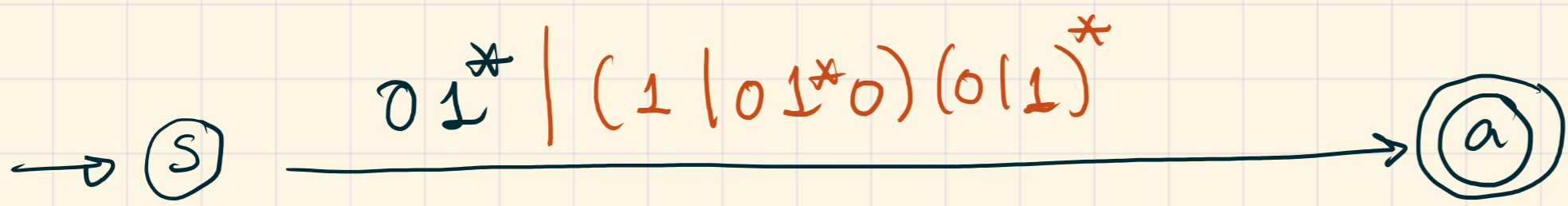
② Delete internal states

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 .



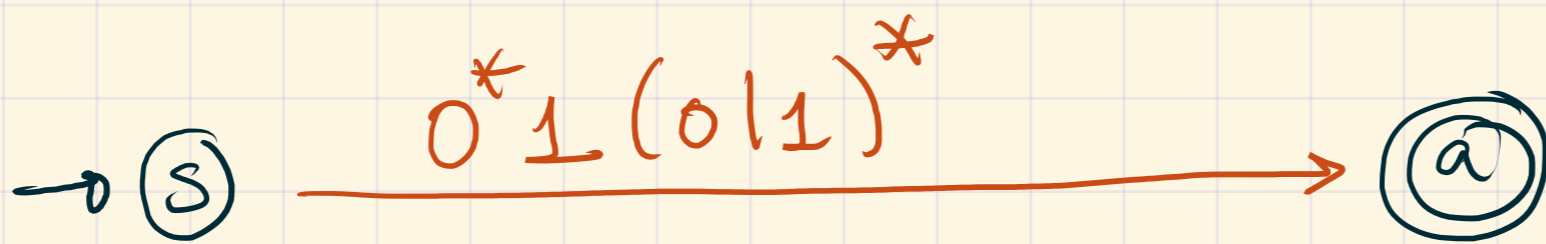
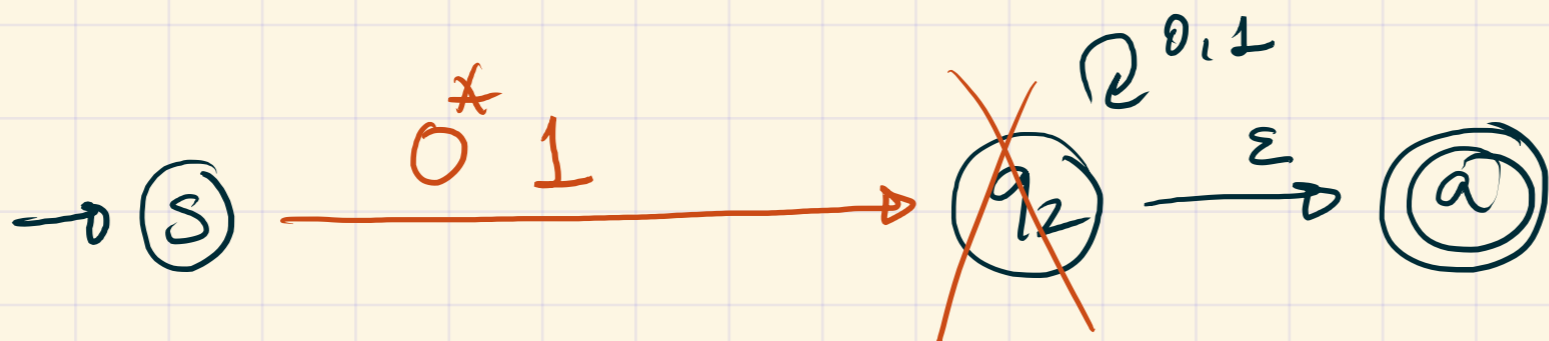
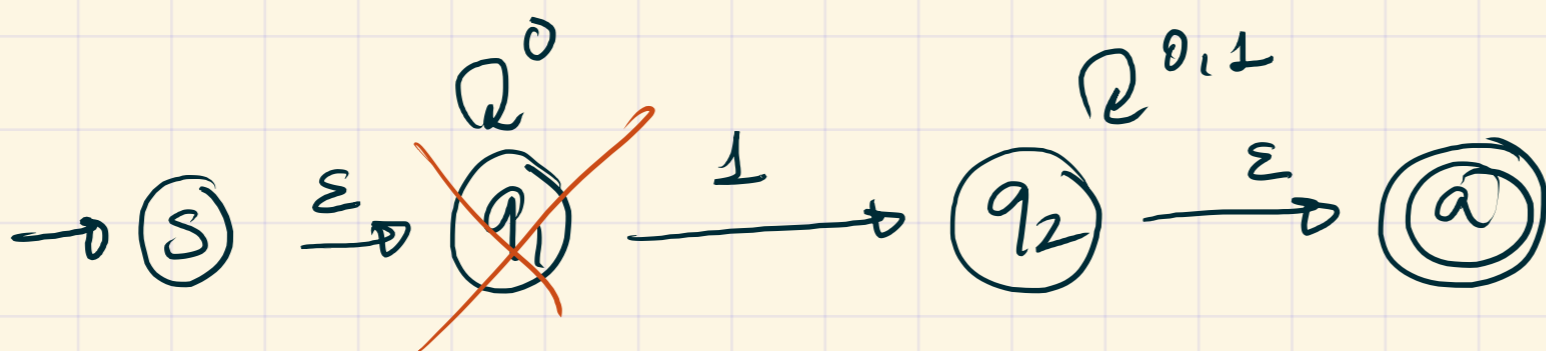
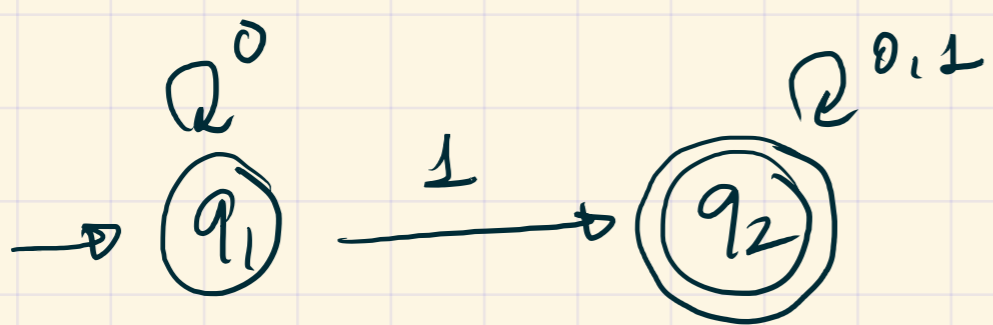


Now we're done, because all internal vertices are gone.

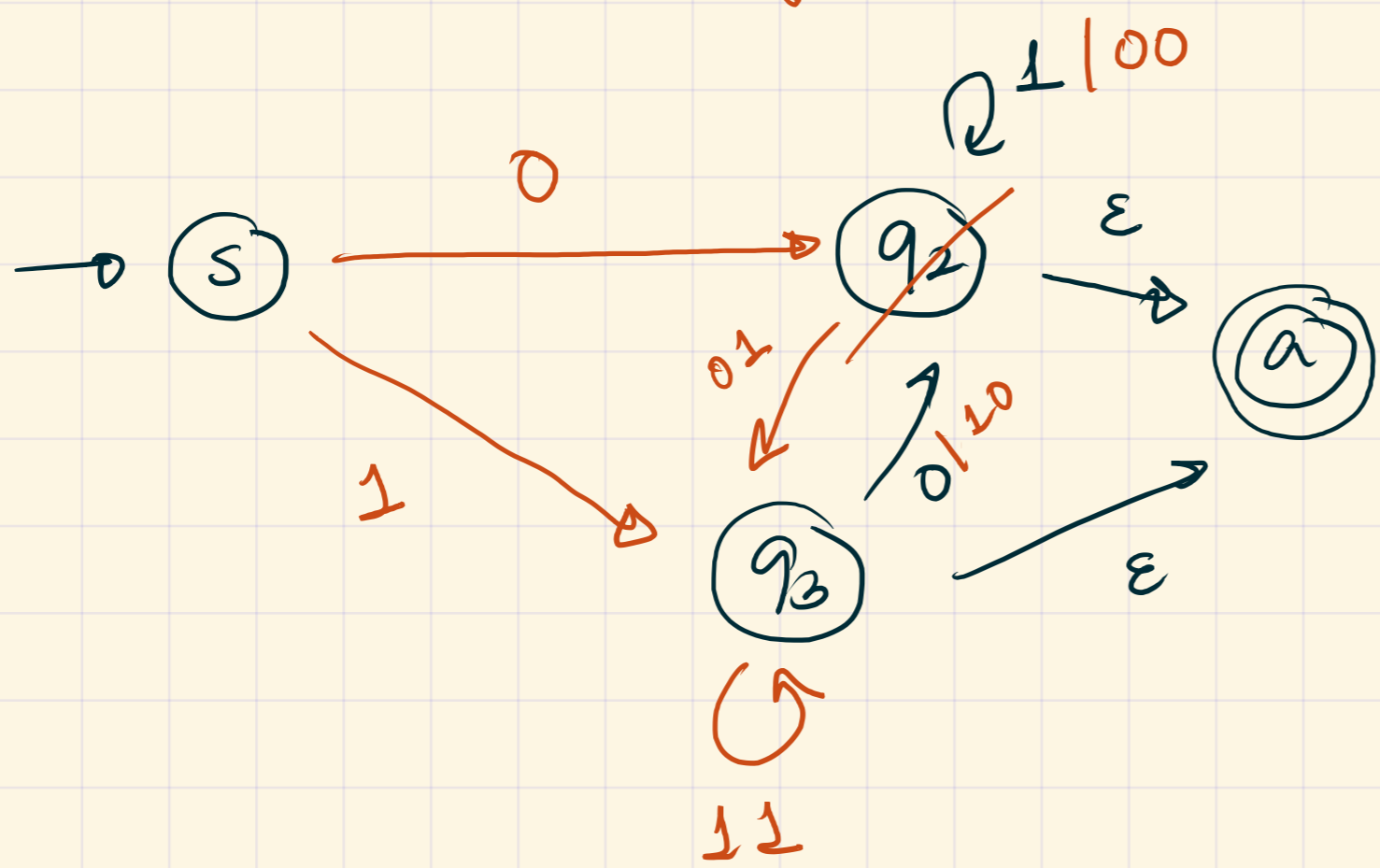
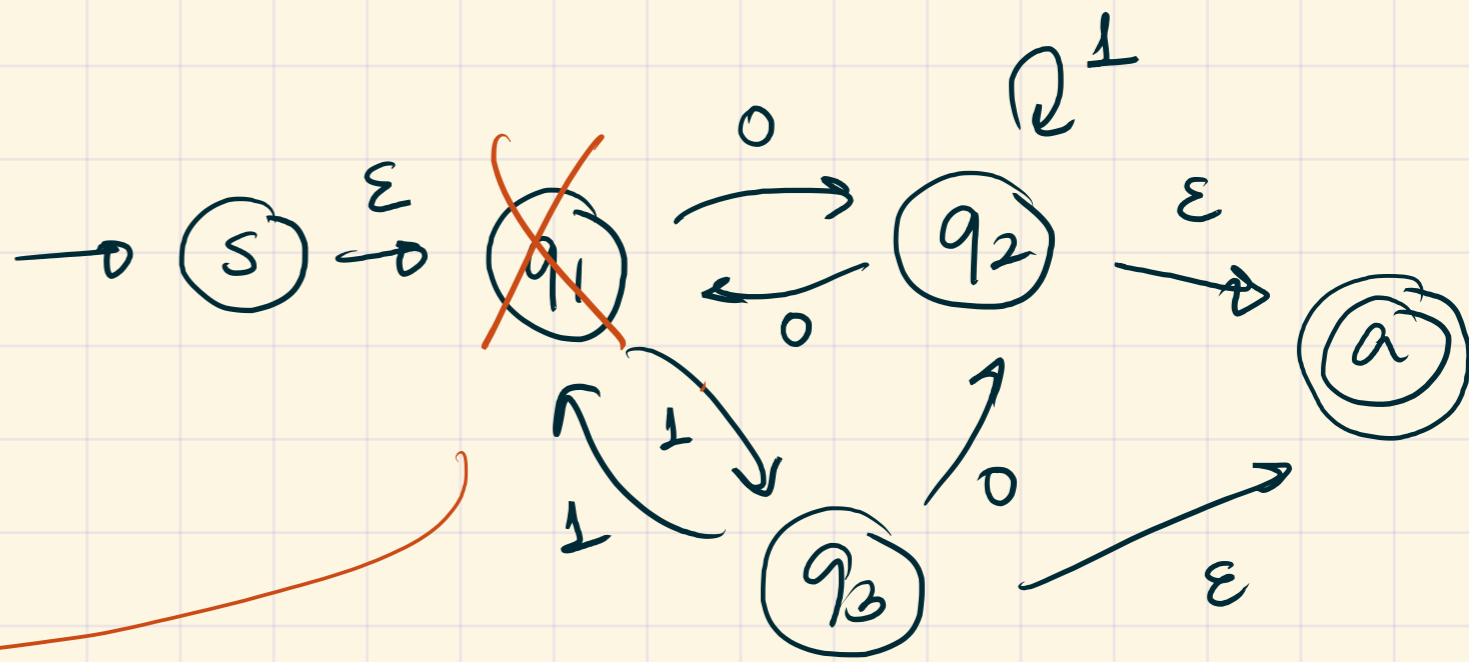
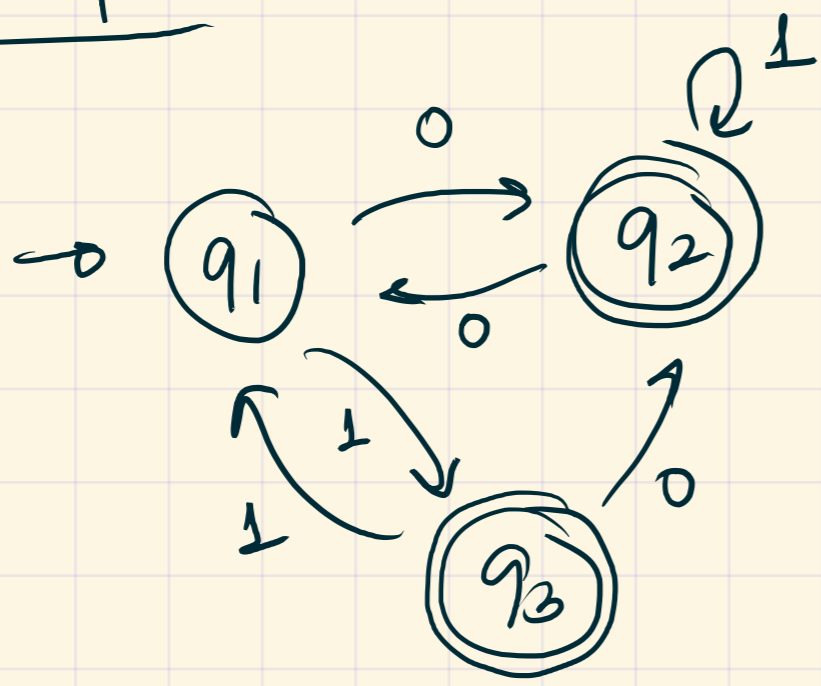
⇒ A regular expression that's equivalent to the original DFA is

$$01^* | (1|01^*0)(011)^*$$

More examples:



Example:



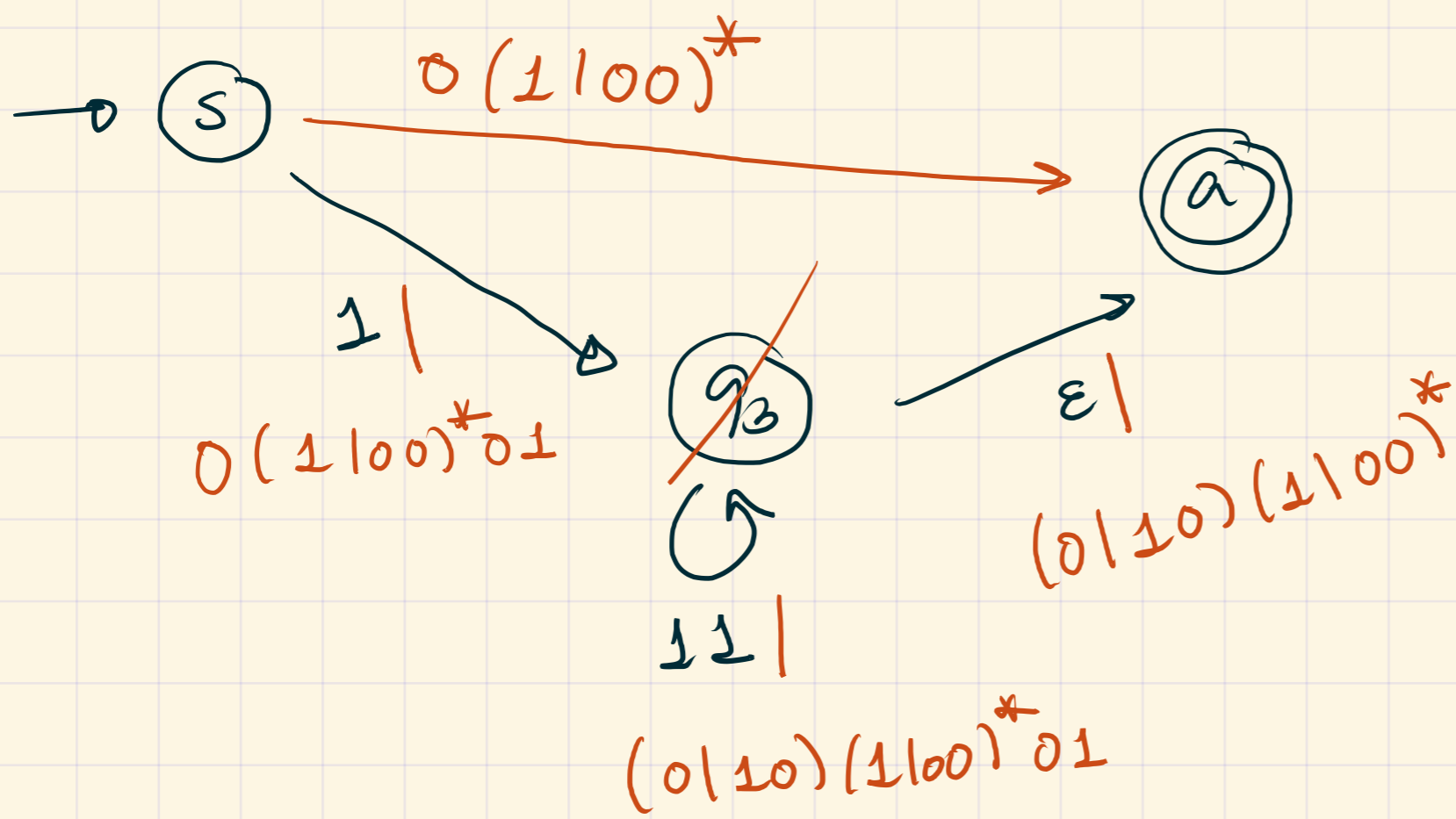
Incoming edges:

- $S \xrightarrow{\epsilon} q_1$ ✓
- $q_2 \xrightarrow{0} q_1$ ✓
- $q_3 \xrightarrow{1} q_1$

Outgoing:

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

[6 combinations]



incoming @ q2:

- $S \xrightarrow{0} q_2$ ✓
- $q_3 \xrightarrow{01} q_2$

outgoing:

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

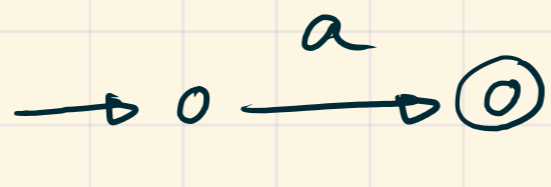
$$\rightarrow S \quad \underline{0(1100)^*} \mid (1 \mid 0(1100)^*01) (11 \mid 010)(1100)^*01 \mid (\epsilon \mid 010)(1100)^*$$

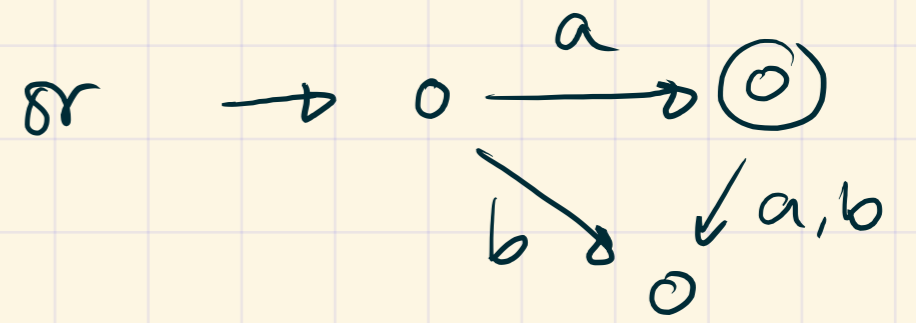
That's the final answer!

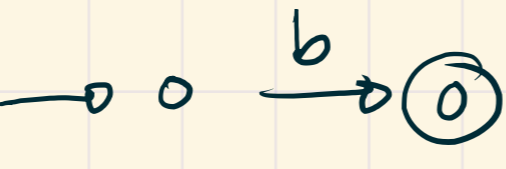
Examples for converting regular expressions into NFAs

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

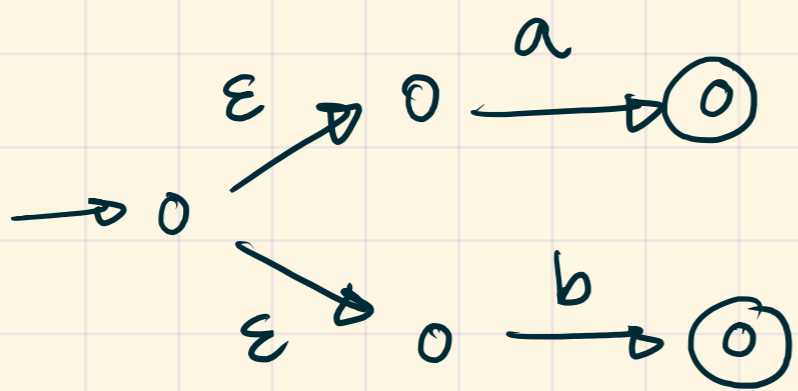
$r = \underbrace{(a|b)^*}_{\text{NFA for } (a|b)^*} \underbrace{aba}_{\text{NFA for } aba}$

NFA for $\{a\}$: 

or 

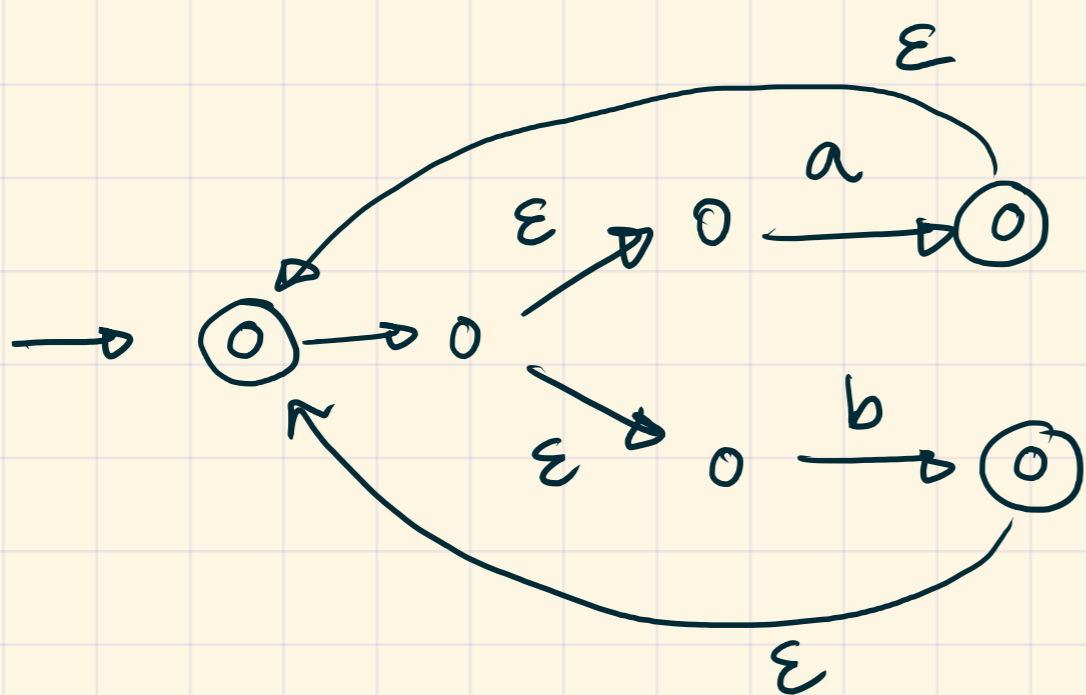
NFA for $\{b\}$: 

NFA for $a|b$



(or product construction of DFAs)

NFA for $(a|b)^*$:



(Finish this next time)