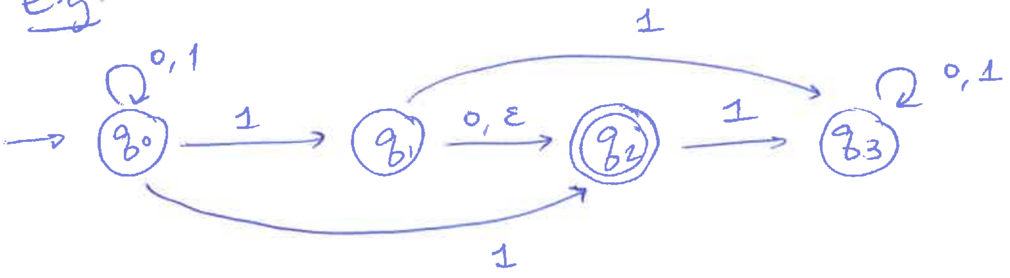
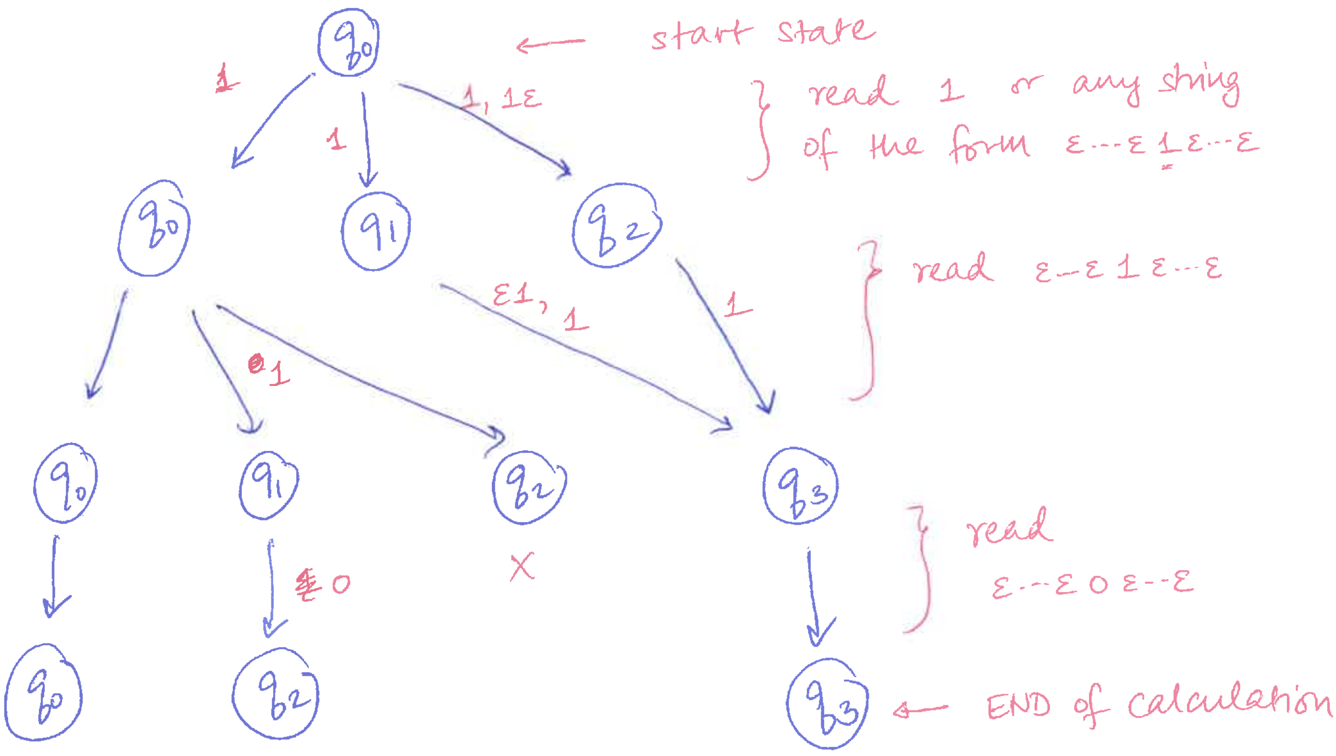
MATH 2301 3/10/2023

* Last time: Nondeterministic finite automata (NFAs)

E.g.



Calculation tree example: $w = 110$



End set of states $= \{q_0, q_2, q_3\}$.

Note: at least one of these is accepting (q_2)

So we accept.

* Rmk :

(1) If in the calculation tree, none of the states on the last level are accepting states, or, if all branches die out before the end of the string, we say that the NFA rejects the string.

(2) If $w = \varepsilon$, we say that an NFA accepts w if there is a path from the start state to some accepting state, exclusively by arrows labelled as ε (s.t each arrow ~~goes~~ in the path has ε as a label.)
Alternatively, if the start state is also an accepting state.

E.g. : In the previous example, $w = 0000$ is not accepted, for example.

Rmk (More later):

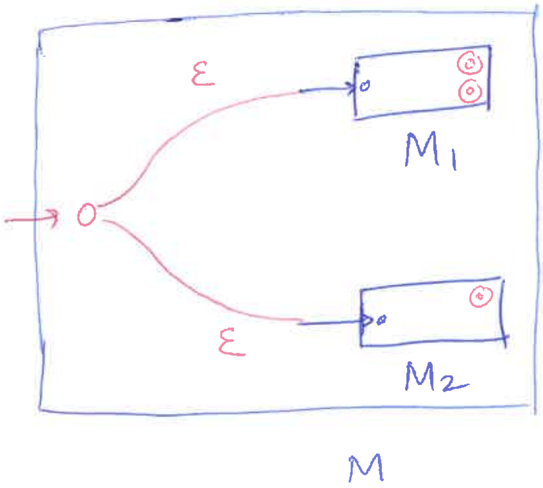
Every NFA that has ε labels can be converted to another NFA that has the same language, but no arrows labelled by ε!

* Regex constructors to NFAs

** Every DFA is also an NFA. So we already have ~~reg~~ conversions from the basic regex constructors ($r = \varepsilon$, $r = a$, $r = \phi$) to DFAs and hence NFAs.

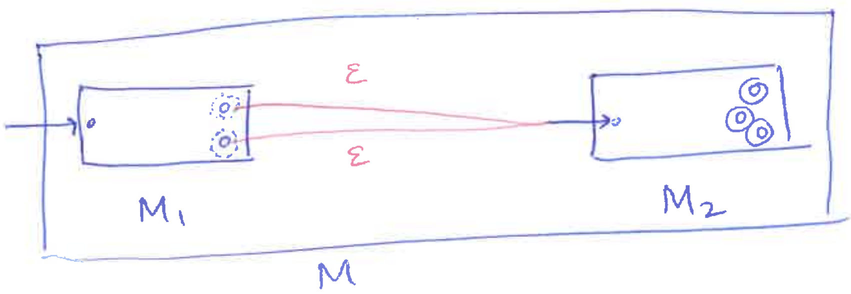
** Back to the other 3 constructors

(4) $r = r_1 \mid r_2$; given machines (NFAs) M_1 & M_2 such that $L(M_1) = L(r_1)$ & $L(M_2) = L(r_2)$, construct M (an NFA) such that $L(M) = L(M_1) \cup L(M_2)$



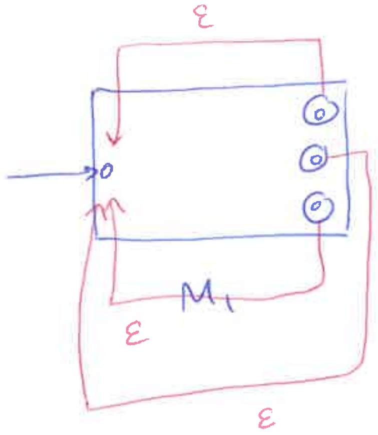
The calculation tree on M runs the possibilities of M_1 and of M_2 simultaneously, accepting if at least one of them accepts.

(5) $r = r_1 r_2$. As before, we have M_1 & M_2 with $L(M_i) = L(r_i)$.



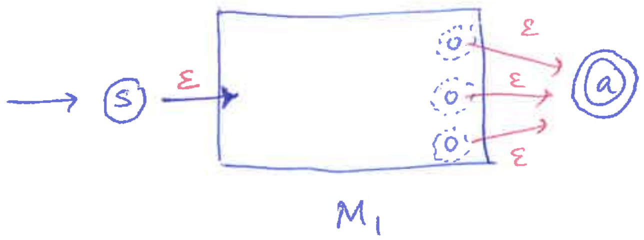
Connected the accept states of M_1 to the start state of M_2 by ε-arrows, and then made them not accepting.

(6) $r = (r_1)^*$, suppose we have M_1 such that $L(r_1) = L(M_1)$.



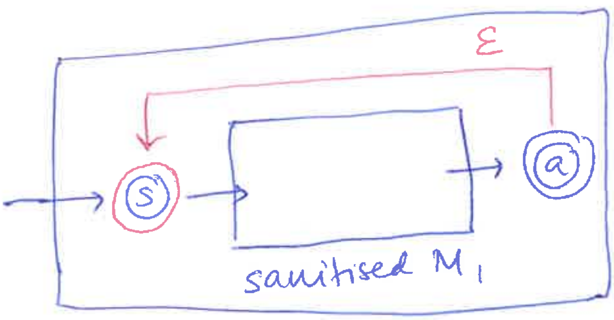
Proposed solution (may not work: this may not recognise ε!)

First build a "sanitised version" of M_1:



Has only one accepting state ⓐ and a separate start state ⓢ, but has the same language as M_1

Build M as follows:



M

M accepts ε, and accepts any concatenation of strings that are each accepted by M_1

$\boxed{\text{UPSHOT}}$: Every regex can be converted to an equivalent NFA, ie, one with the same language.