MATH 2301

29/09/2023

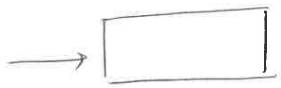
* Admin : Next lecture Tuesday 03 October
2pm – 3pm in Robertson (same room as Wed lecture)

* Last time : Tried to construct DFAs to mimic regex constructors (we mostly failed for ④, ⑤)

* Today : Alternate way to (partially) succeed ~~@~~, and a new flavour of machines.

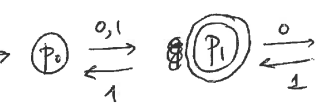
④ $r = r_1 \mid r_2$; $L(r) = L(r_1) \cup L(r_2)$.

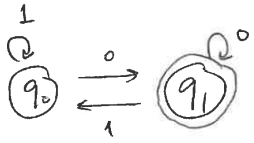
Given M_1, M_2 with $L(M_i) = L(r_i)$,
construct M s.t. $L(M) = L(r_1) \cup L(r_2)$
$= L(M_1) \cup L(M_2)$



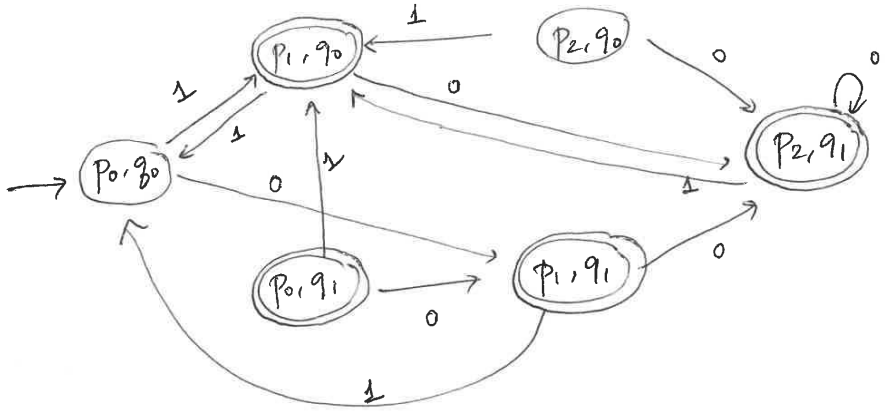
Given w, simulate M_1 and M_2 on w simultaneously?

* Solution : Take the "product" DFA :

E.g. $M_1 =$ 

$M_2 =$ 

Product automaton : (Keeps track of states in M_1 & M_2 simultaneously)



← M

$L(M) = L(M_1) \cup L(M_2)$

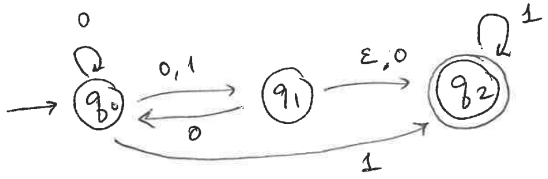
Accept states are (x, y) where either x is accepting in M_1 or y is accepting in M_2 (or both).

* What about concatenation and star??

* The other constructors are hard to construct DFAs for! We'll introduce an upgrade to our machines to make it easier.

* Non-deterministic finite automata (NFAS)

Example



An NFA state diagram resembles a DFA state diagram, with the following changes:

(1) You can have states with no outgoing arrows labelled by one or more letters.

(2) You can have multiple outgoing arrows labelled by the same letter.

(3) You can have ε as edge labels.

Def: An NFA consists of the following:

(1) An alphabet Σ

(2) A set of states Q

(3) A start state $q_0 \in Q$

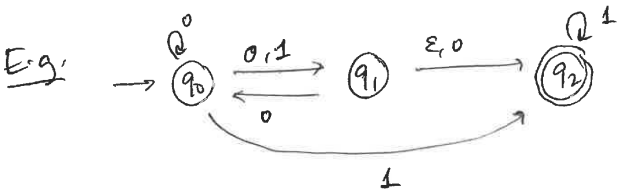
(4) A set of accept states $A \subseteq Q$

(5) A transition function

$$\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \longrightarrow \mathcal{P}(Q)$$

Q: state you're at; $\Sigma \cup \{\varepsilon\}$: what you read; $\mathcal{P}(Q)$: where you go (a set of states)

Eg:

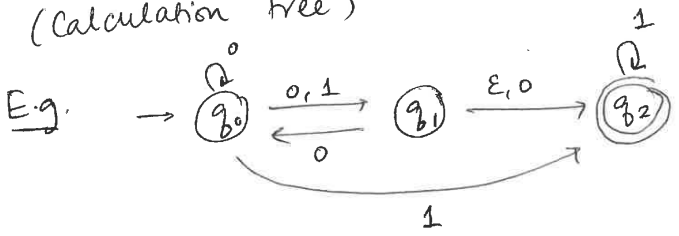


$\Delta(q_0, 1) = \{q_1, q_2\}$ $\quad \Delta(q_0, \varepsilon) = \emptyset$

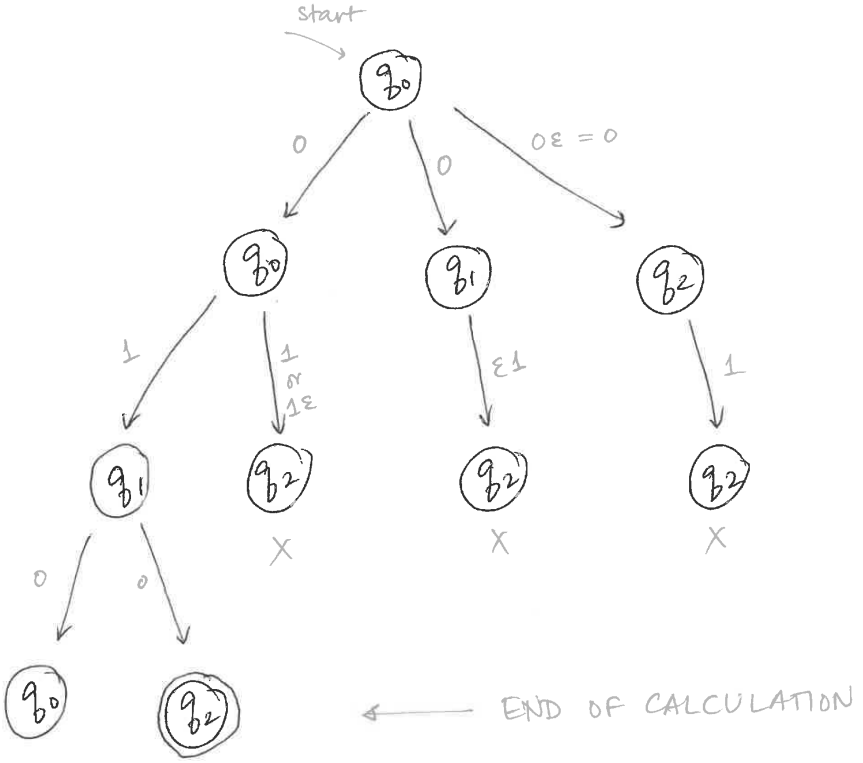
$\Delta(q_0, 0) = \{q_0, q_1\}$ $\quad \Delta(q_1, 1) = \emptyset$

* How to compute with an NFA?
(Calculation tree)

E.g.



E.g string
w = 0 1 0



Final set of states is $\{q_0, q_2\}$; q_2 accepts so we accept.

(one of the final set is accepting!)