MATH 2301 — 06 Oct 2022

* Rmk about calculation tree for NFA

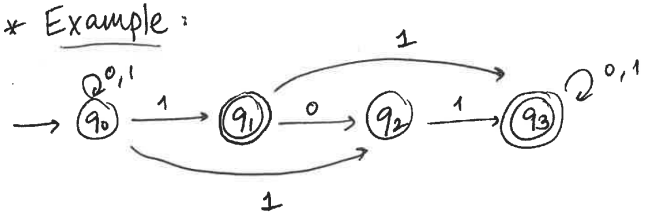
If $w = \varepsilon$, and M is an NFA then w is accepted by M if and only if there is a path from the start state to some accept state, which follows only ε-labelled arrows.

---

* Yesterday + Today: Convert NFAs to equivalent DFAs

- Step 1: Eliminate any ε-arrows
- Step 2: (Main step): Construct a DFA whose state set is the power set of the original set of states; the transition function is read off using the same process as in the calculation tree.

* Example:



(NFA without ε-arrows)

M

Let's construct a DFA M', which is equivalent.

- State set of $M' = \mathcal{P}(Q)$, where $Q = \{q_0, q_1, q_2, q_3\}$
- Start state of $M' = \{q_0\}$
- Accepting states of $M' = \{$any subset of of Q that ~~cont~~ contains any of the accepting states of $M'\}$

E.g (in our example): $\{q_1\}$, $\{q_3\}$, $\{q_1, q_3\}$, $\{q_0, q_1\}$, $\{q_0, q_3\}$, $\{q_0, q_1, q_2, q_3\}$, ... etc

Non-accepting: $\emptyset$, $\{q_0\}$, $\{q_2\}$, $\{q_0, q_2\}$.

- Transition function:

$\delta' : \mathcal{P}(Q) \times \Sigma \longrightarrow \mathcal{P}(Q)$

[Let $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ be the transition fn of M.]

~~$\delta'(\not{A}, x) = \not{\in}(\not{x})$~~

a

Let $B \in \mathcal{P}(Q)$, i.e. $B \subseteq Q$. Let $x \in \Sigma$.

$$\delta'(B, x) = \bigcup_{b \in B} \delta(b, x)$$

[Read x at each state in B, and combine all outputs together.]

- Resulting machine M' is a DFA ✓
- ~~Eta~~ Can check that w is accepted by M if and only if w is also accepted by M'.

[Running a word w through M' is the same process as running the calculation tree of w on M.]

⇒ Every NFA has an equivalent DFA!

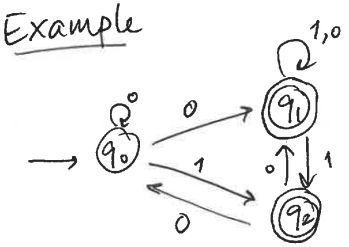
⇒ Each regex has an equivalent ~~DF~~ NFA and hence an equivalent DFA!

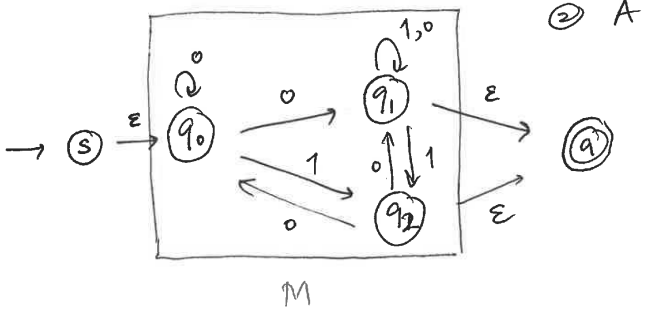
[Downside: NFA → DFA construction gives an exponential blow-up in terms of size.]

---

* Next goal: Go backwards [NFA/DFA → regex]

Example





M

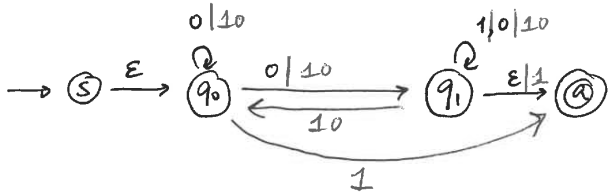
Pre-step: clean up

① Make a new start state ⓢ, with an ε-arrow to the original start

② A new accept state ⓐ, and ε-arrows from all old accept states to ⓐ.

Main algorithm: Successively remove one state at a time from inside the box, updating arrows as you go.

E.g.: Let us remove q_2 (+ all arrows to/from q_2)



[Generalised NFA]

[Continue on Monday!]