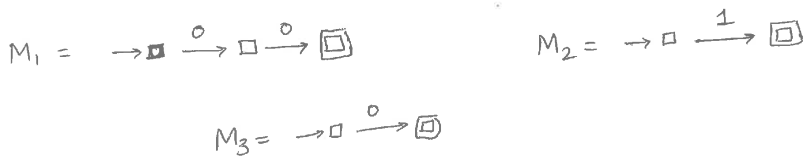
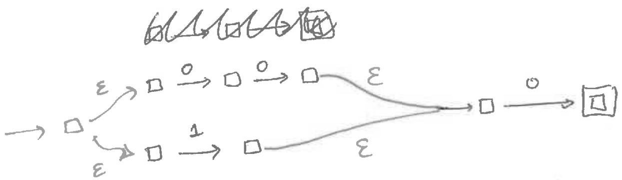
MATH 2301 — 04/10/2023

* <u>Last time</u>: Converted all regex constructors to NFAs.

<u>Upshot</u>: We can combine these building blocks to generate an NFA recognising the same language as any given regex!

<u>Example</u>: $r = \underbrace{(00}_{r_1} | \underbrace{1}_{r_2}) \underbrace{0}_{r_3}$





M

$L(M) = L(r)$.

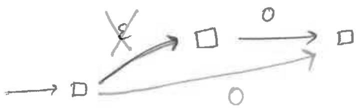
(I drew squares for states because circles look like zeroes)

* <u>Today</u>: DFAs vs NFAs

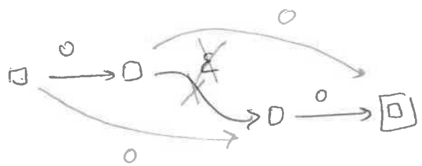
NFAs are clearly more powerful than DFAs.
Are they strictly more powerful?

<u>Q</u>: Given an NFA M, how much/how closely can you simulate it using a DFA?

* <u>Scratch Work</u>



Zoom in on 1^{st} ε-arrow
How to remove it?
(without messing up calculation)



another part of machine

We were able to delete some ε-labels at the cost of adding extra transitions labelled by letters.

<u>Proposition</u>: Given any NFA M that possibly has ε-labels, there always exists an equivalent NFA M' which has <u>no</u> ε-labels.
(Equivalent means that $L(M) = L(M')$.)

<u>Pf</u>: Skipped, but essentially it is what we did in scratch work, but more systematically.

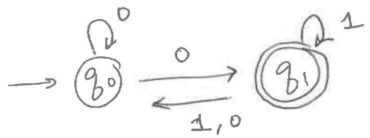
* Simplifying NFAs : remove ε-transitions.

(We just stated that we can do this)

---

* Now suppose we have an NFA that has no ε-labels.

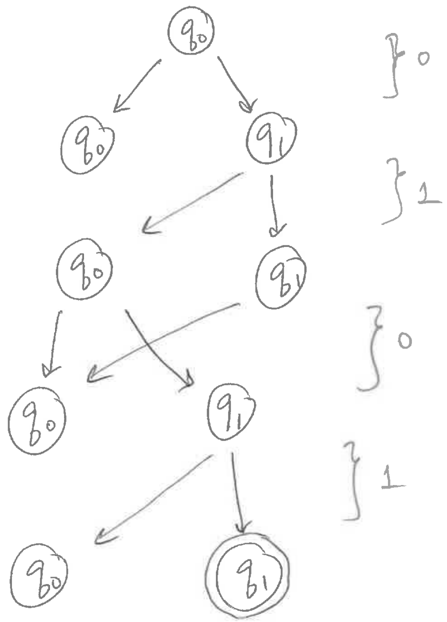
E.g.



Not a DFA.

Problem : We can have zero or more outgoing arrows labelled by a given letter.

Look at calculation tree $w = 0101$

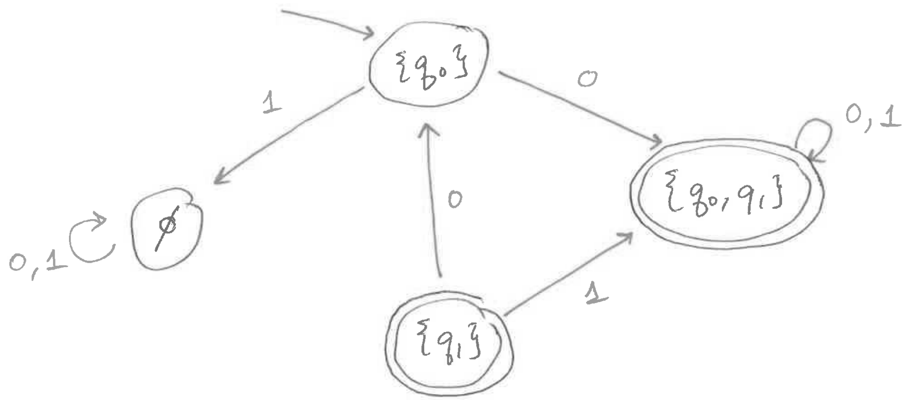


After each step of the calculation tree, we reach some subset of Q.

Let us construct a "power-set" automaton.

Original NFA had Q as the state set.

Draw a new machine whose states are all possible subsets of Q.



This is a DFA, which does the same thing as the previous NFA.

E.g. $w = 0101$ (compare this machine w/ previous NFA)

* Summary

Consider an NFA M, suppose it has no ε-arrows.

We construct a new DFA M' as follows:

- State set of M' is $\mathcal{P}(Q)$, where Q = state set of M.
- Start state of M' is $\{q_0\}$
- Accepting states of M' are all subsets of Q that contain at least one accepting state of Q.

- Transition function:

Given $X \subseteq Q$ and a letter $a \in \Sigma$

$$\delta(X, a) = \bigcup_{q \in X} \Delta(q, a)$$

(for M')

transition fn of M

Apply the letter a to every state q of M that lies in X, and combine the outputs by a union.