MATH 2301

04/10/2023

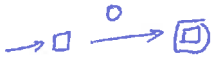
* Last time: Converted all regex constructors to NFAs.

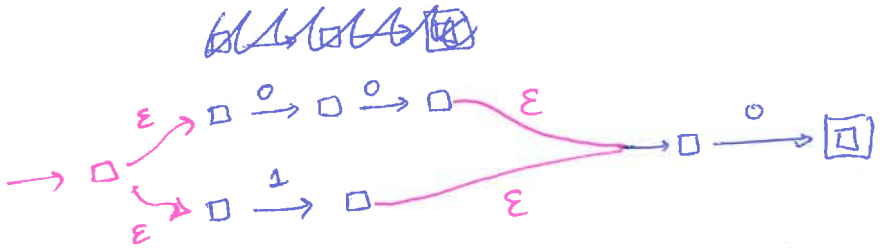
Upshot: We can combine these building blocks to generate an NFA recognising the same language as any given regex!

Example: $r = (00 \mid 1)0$ (r_1 r_2 r_3)

$M_1 =$ 

$M_2 =$ 

$M_3 =$ 



M

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

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

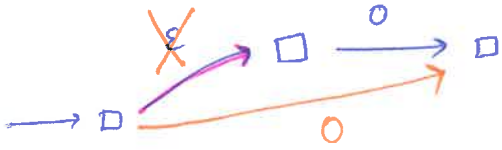
---

* Today: DFAs vs NFAs

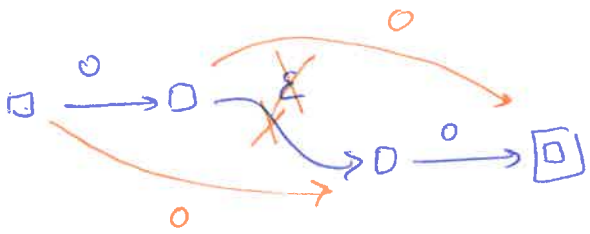
NFAs are clearly more powerful than DFAs.
Are they strictly more powerful?

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

* Scratch Work



Zoom in on 1st 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.

Proposition: Given any NFA M that possibly has ε-labels, there always exists an equivalent NFA M' which has no ε-labels.

(Equivalent means that $L(M) = L(M')$.)

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

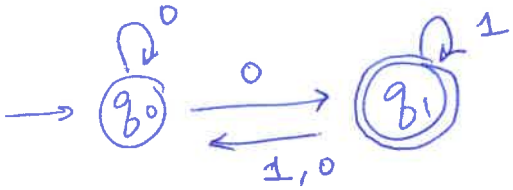
* Simplifying NFAs : remove ε-transitions.

(We just stated that we can do this.)

---

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

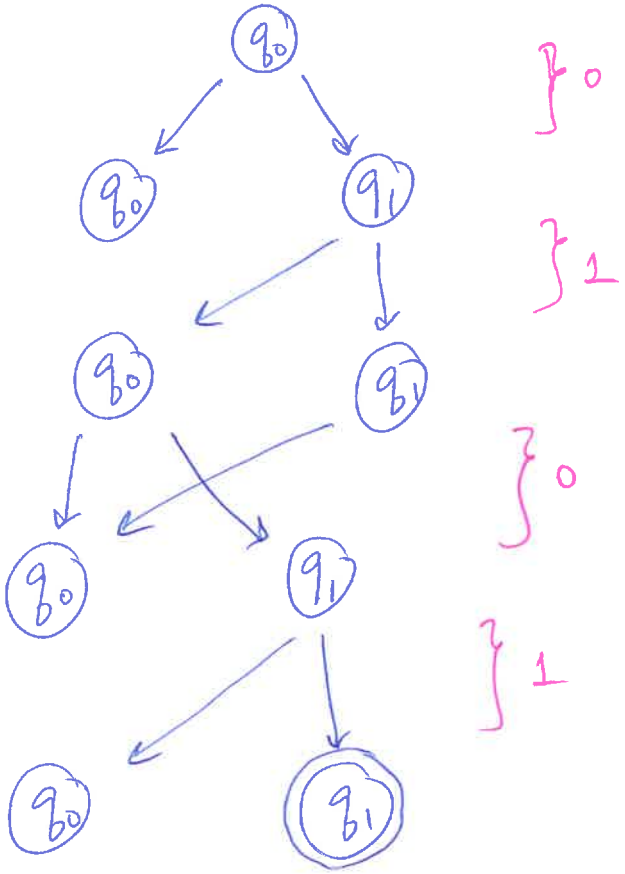
E.g.



Not a DFA.

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

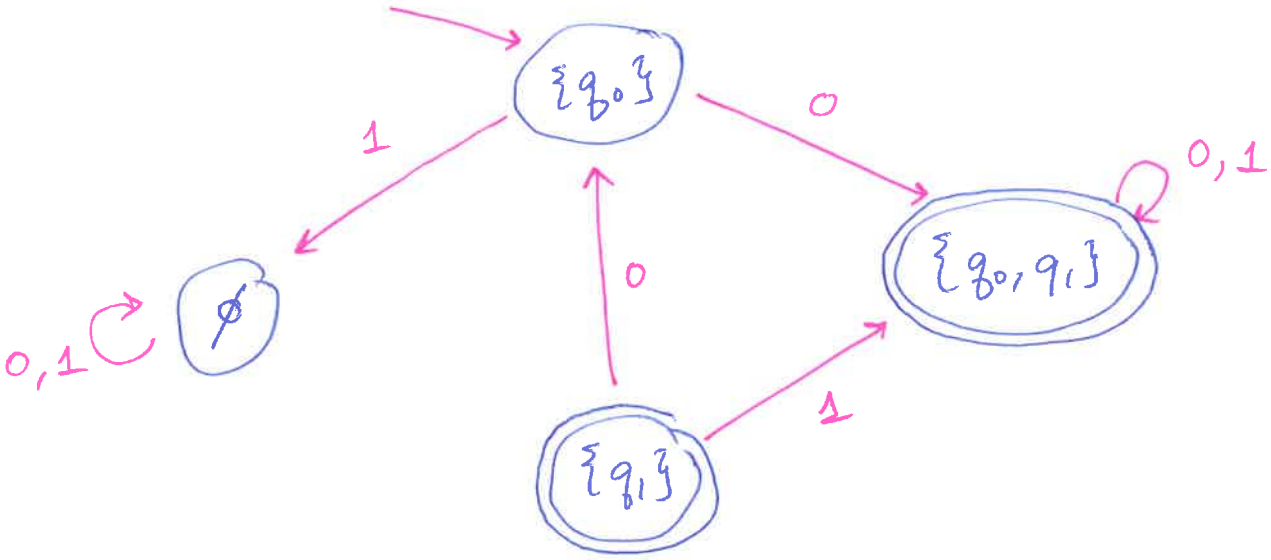
Look at calculation tree $w = 0\,1\,0\,1$



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')

Δ: 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.