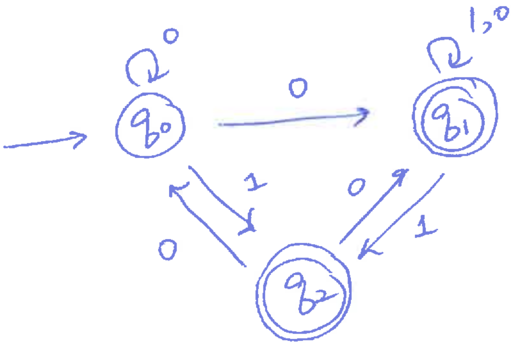
MATH 2301

09/10/2023

* **Recap**: Given an NFA M, can we construct a regex r such that

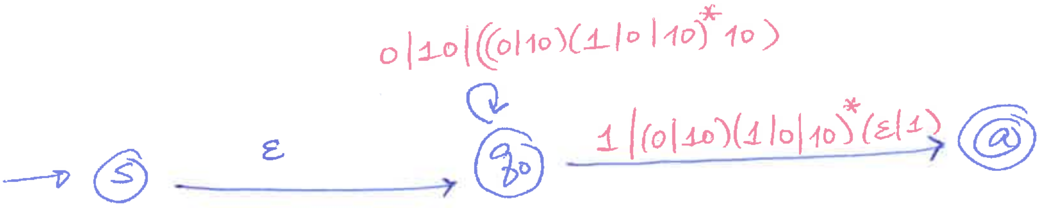
$L(M) = L(r)$?

* **Last time**: Described a procedure for this.



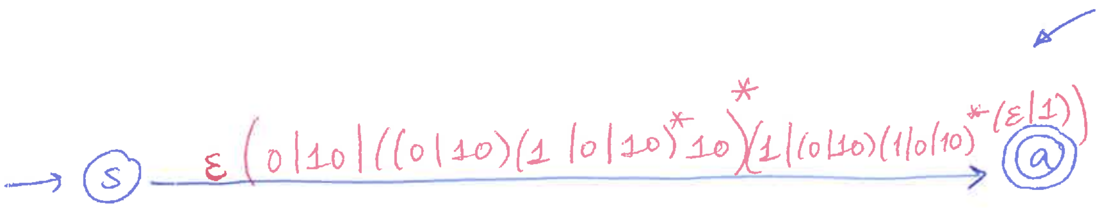
(original machine M)

We converted it to a generalised NFA (GNFA):



Last step: Remove q_0

Call the final label r



By construction, $L(M) = L(r)$!

* Summary

Step 0 : Sanitise M by pulling out a separate start state ⓢ and a separate (single) accept state ⓐ, redirecting all previous accepting states to ⓐ via ε-arrows, and ⓢ to the previous start state via an ε-arrow.

Step 1 : Choose some state ⓠ (different from ⓢ or ⓐ) to delete, and delete it.

For each pair x, y such that

$$\textcircled{x} \xrightarrow{\ell_1} \overset{\ell_2}{\circlearrowleft}\textcircled{q} \xrightarrow{\ell_3} \textcircled{y}$$

were previous labels,

{ x can be equal to y, but $x \neq q$ and $y \neq q$

add an arrow $\textcircled{x} \longrightarrow \textcircled{y}$, labelled by

$$\ell_1 (\ell_2^*) \ell_3 .$$

[If there is no loop ℓ_2, just drop it to make $\ell_1 \ell_3$.]

Step 2 : Repeat step 1 until there is no other state left, other than ⓢ and ⓐ.

Let r = label on the arrow $\textcircled{s} \longrightarrow \textcircled{a}$.

That is your answer!

When adding $\ell_1 (\ell_2^*) \ell_3$ to $\textcircled{x} \longrightarrow \textcircled{y}$, if there is already some label ℓ on $\textcircled{x} \longrightarrow \textcircled{y}$, the new label becomes

$$\ell \mid \ell_1 (\ell_2^*) \ell_3$$

(Two options on one label get joined by |)

* **Def:** We say that a language L is **regular** if: one of the following (equivalent) conditions holds:

(1) There is a DFA M such that $L(M) = L$,

(2) There is an NFA M such that $L(M) = L$,

(3) There is a regex r such that $L(r) = L$.

(Our discussion so far has shown that any one of these implies the other two.)

Q: Does this mean that some languages are not regular??

E.g.: $L = \{0^n 1^n \mid n \in \mathbb{N}\}$

$L = \{\varepsilon, 01, 0011, 000111, \dots\}$

I claim that L is not regular.

An answer: countability. [ASIDE]

Each regex is finite in size. (finitely many symbols in r)

One could enumerate all possible regexes in dictionary order: $r_1, r_2, r_3, r_4, \dots$

Consider $L(r_1), L(r_2), L(r_3), \dots$ ← List of all possible regular languages (with repeats)

Now on the other hand, the set of all possible languages is uncountable. [NOT LISTABLE]

$\Rightarrow$ There are non-regular languages

Beyond the scope of this class

Back to

$L = \{ 0^n 1^n \mid n \in \mathbb{N} \}$ ← This description is <u>not</u> a regex.

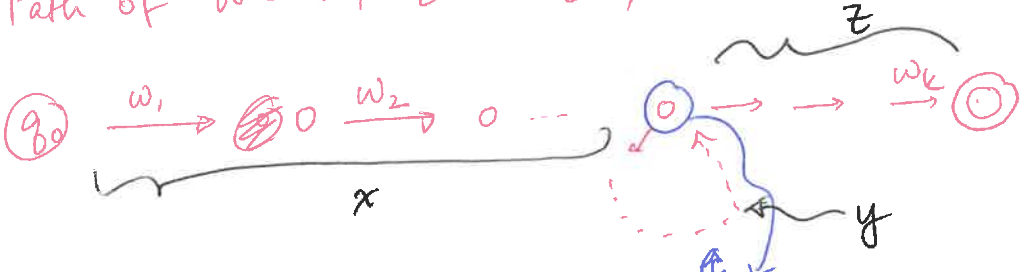
<u>Claim</u>: There is no DFA M such that $L(M) = L$.

<u>Pf</u>. Suppose there existed some DFA M such that $L(M) = L$.

M has some number of states, say N states.

If w ~~does~~ is long enough, i.e. $|w| > N$ then as w runs through M, it must repeat a state.

Path of $w = w_1 w_2 \cdots w_k$; say w accepted by M



This state repeated.

w can be split into three parts $x\,y\,z = w$

Note that $x\,yy\,z$, $x\,yyy\,z$, $x\,yyyyyyy\,z$ are also accepted by this DFA M.

⑤

M has N states,

$L = \{0^n 1^n \mid n \in \mathbb{N}\}$.

Q: What happens if you feed the string $w = 0^N 1^N$ into M?

→ Can you produce a contradiction from this?