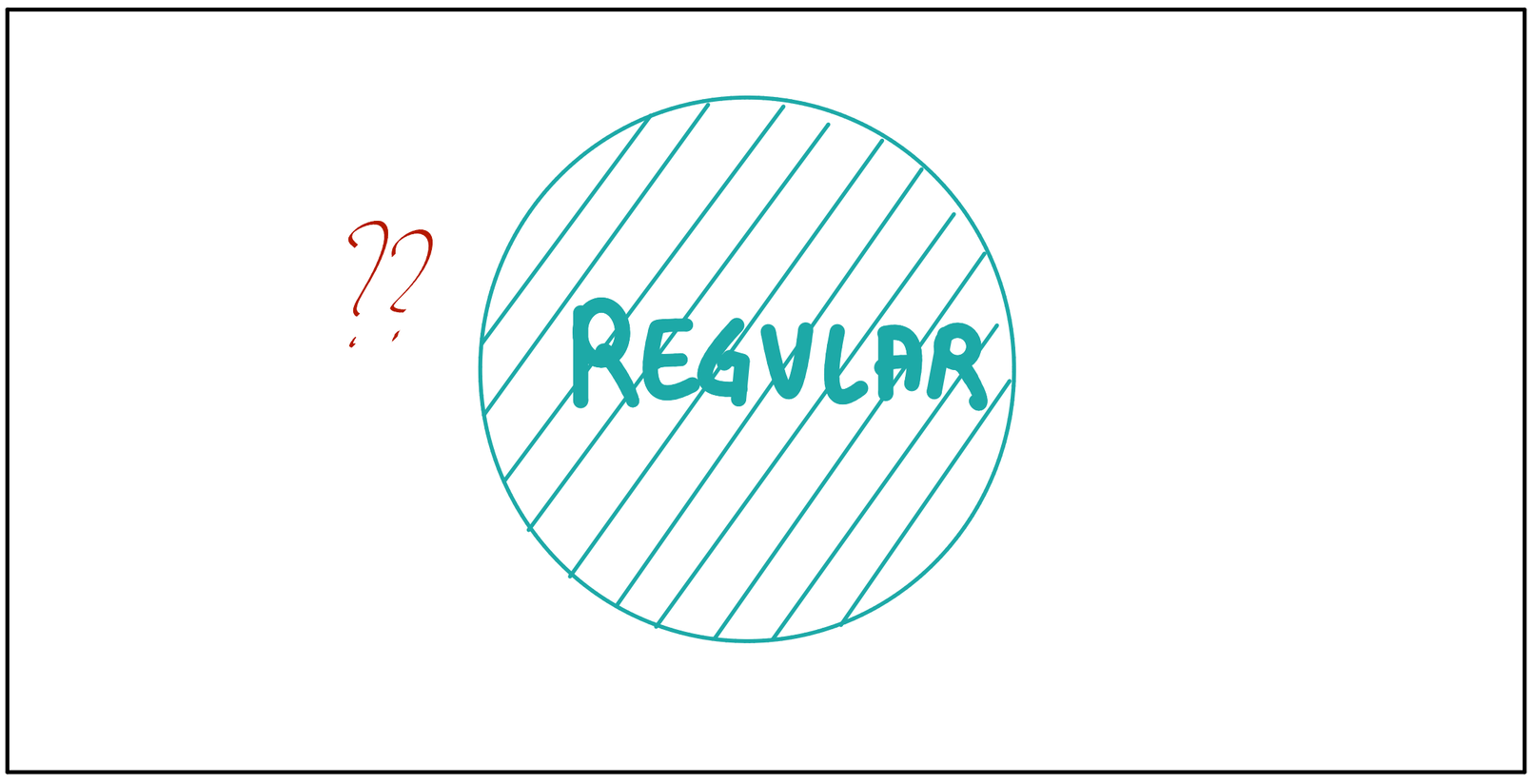
Limitations Of DFAs

We saw many examples of regular languages

Also saw equivalent representations by way of DFA/NFA/regex.

Today we will look at the limitations of DFAs.



We will look at a variant of problem 14 from the tutorial sheet.

If L is regular, so is $L_f = \{x \mid x \cdot x \in L\}$

We have a DFA M for L.

Somehow we want to extract an NFA for L_f from M.

A general strategy for such problems is to think of flags on the states of M. What config do you start with? How do the flags move wrt transitions? Where are the flags situated in accepting configs?

$$L_f = \{x \mid x \cdot x \in L\}$$

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing L.

Put a green flag on q_0, and guess some state q, where we put two flags — one blue, and one red.

Never move the red flag.

Move the green and blue flags in lock step, according to the input letter.

When do we accept?

Can we now formally describe the NFA M' for α_f?

$$M' = (Q', \Sigma, \Delta, Q_0, F')$$

$$Q' = Q \times Q \times Q$$

green, red, blue

$((q_1, q_2, q_3), a, (q_1', q_2', q_3')) \in \Delta$ iff

$q_1' = \delta(q_1, a)$, $q_2' = \delta(q_2, a)$, $q_3' = q_3$, for $q_1, q_2, q_3 \in Q$, $a \in \Sigma$

$$Q_0 = \{(q_0, q, q) \mid q \in Q\}$$

$$F' = \{(q, f, q) \mid q \in Q, f \in F\}$$

Prove that $\alpha_f = \alpha(M')$.

So we showed that if $\mathcal{L}$ is regular, then $\{x \mid x \cdot x \in \mathcal{L}\}$ is regular.

Is the reverse true?

Is $\mathcal{L} = \{x \cdot x \mid x \in \mathcal{L}_f\}$ regular, if $\mathcal{L}_f$ is regular?

Can I perform a Concat-like construction?

Hard to "figure out where to break the string" in advance"

Suppose I add an actual separating character.

Is $\mathcal{L}_{new} = \{x \diamond x \mid x \in \mathcal{L}_f\}$ regular, where $\mathcal{L}_f \subseteq \Sigma^*$ and $\mathcal{L}_{new} \subseteq \Sigma^* \diamond \Sigma^*$?

What about $\mathcal{L}_k = \{x \cdot x \mid x \in \mathcal{L}_f, |x| \leq k\}$?

Does this help us narrow down the issue?

The problem is to do with x being unboundedly long.

How much can a DFA remember?

Essentially, just a state and a letter.

Keeping track of an unbounded string with no set pattern
in order to match it against "future" letters is beyond a DFA!