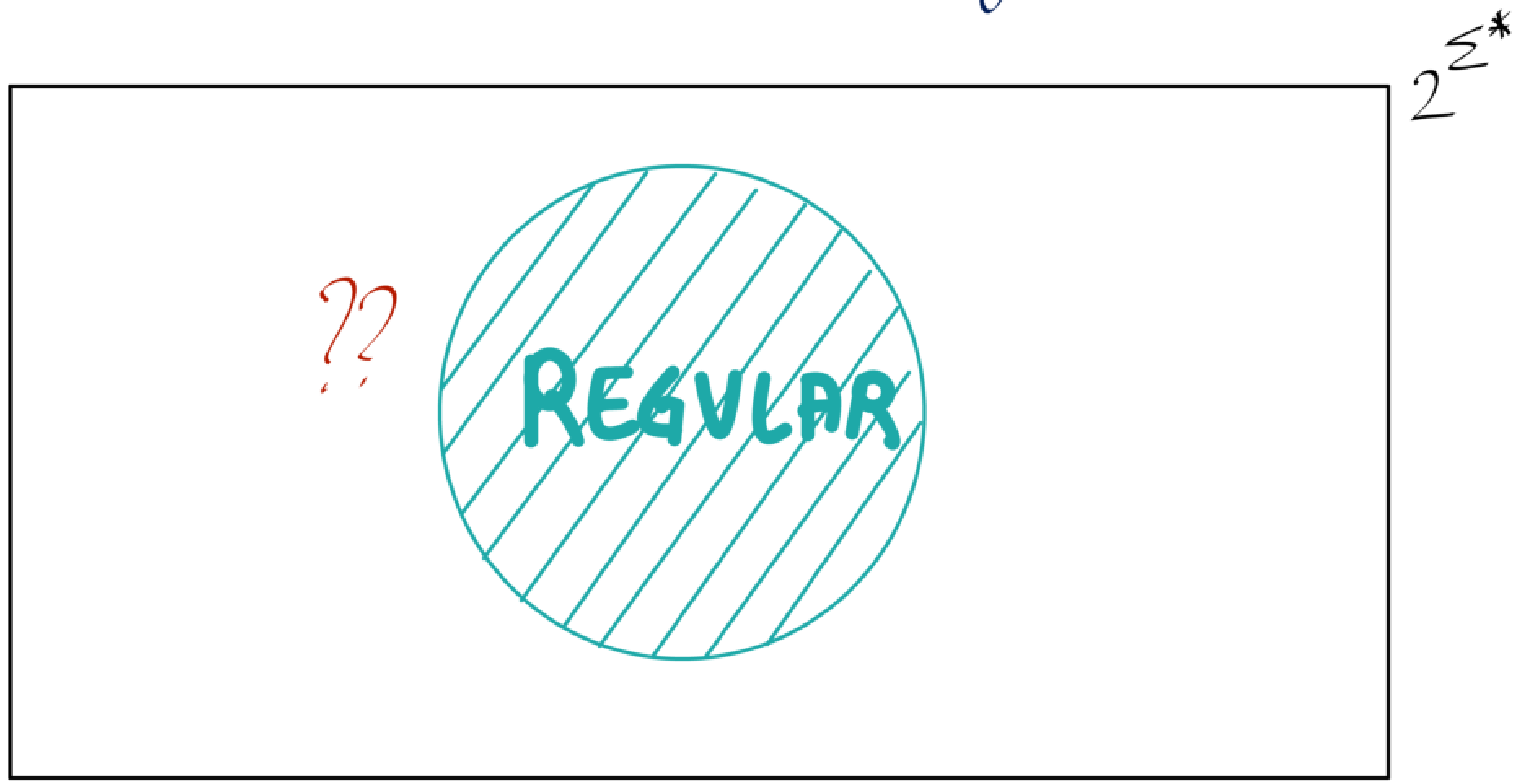
Pumping Lemma

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.



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 $\mathcal{L}_{new} \subseteq \Sigma^* \diamond \Sigma^*$?

The problem is to do with x being unboundedly long

with no finite repeating pattern

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!

It is hard to "match" two unboundedly long halves of a string.

The canonical example for illustrating non-regularity is

$$\mathcal{L} = \{a^n b^n \mid n \geq 0\} \subseteq \{a, b\}^*.$$

Recall the Myhill-Nerode theorem.

We defined $\sim_L \subseteq \Sigma^* \times \Sigma^*$ as follows:

$x \sim_L y$ iff for every $z \in \Sigma^*$, $xz \in L$ iff $yz \in L$.

The theorem said the following:

L is regular iff $\sim_L$ induces finitely many equivalence classes.

$L = \{a^n b^n \mid n \geq 0\}$. Consider $a^i, a^j \in \Sigma^*$ for $i \neq j$. Is $a^i \sim_L a^j$?

One can also present a more "machine-centric" view of non-regularity.

Suppose there was a DFA $M = (Q, \Sigma, \delta, q_0, F)$ which could recognize $\mathcal{L}$.

It has to have a finite number of states. Suppose $|Q| = 10$.

Consider a string $\omega = a^{15} b^{15}$. $\omega \in \mathcal{L}$, obviously.

So there must be an accepting run of M on ω, i.e. $\hat{\delta}(q_0, \omega) \in F$

We only have 10 states, and 15 'a's to read.

By the pigeonhole principle, at least one state must be repeated while reading this sequence of 'a's.

Suppose this state is $q \in Q$.

Suppose q is reached after we have read 4 a's, and again after we have read 13 'a's.

We can split up our string as follows:

aaaa aaaaaaaaa aa bbbbbbbbbbbbbbb

q_0 $\quad$ q $\quad$ q $\quad$ $f \in F$

$\hat{\delta}(q_0, a^4) = q \qquad \hat{\delta}(q, a^9) = q \qquad \hat{\delta}(q, a^2 b^{15}) = f \in F$

What does this tell us about the behaviour of M on

$a^6 b^{15}$?

$a^{24} b^{15}$?

We could have repeated this argument for $|Q| = k$ for any k, and for any $a^n b^n$ where $n > k$.

This tells us that DFAs cannot count arbitrarily many characters.

We want to say that for a language $\mathcal{L}$, if

$\forall$ no matter what DFA one presents (with k states),

$\exists$ one can produce a string with a "middle" of length $\geq k$, s.t.

$\forall$ no matter how one chooses to break this "middle" into three parts,

$\exists$ one can either remove, or add extra copies of the second part to obtain a string NOT in $\mathcal{L}$,

then $\mathcal{L}$ is not regular.

Formally, we state the Pumping Lemma as follows.

Consider a language $\mathcal{L} \subseteq \Sigma^*$. If

$\forall$ for any $k \geq 0$,

$\exists$ there is an $xyz \in \mathcal{L}$ with $|y| \geq k$, and

$\forall$ for any $u, v, w \in \Sigma^*$ s.t. $uvw = y$ and $|v| > 0$,

$\exists$ there is some $i \geq 0$ s.t. $xuv^iwz \notin \mathcal{L}$, then

$\mathcal{L}$ is not regular.

The nice quantifier alternation allows us to set this up as a game between the $\forall$ player and the $\exists$ player.

The $\forall$ player is the adversary, who believes $\mathcal{L}$ is regular.
The $\exists$ player is the prover, who believes $\mathcal{L}$ is not regular.

The language under consideration is not regular if the $\exists$ player always has a winning strategy.

no matter what the adversary does,
the prover can produce witnesses for the $\exists$ and win!

Strategies will depend on the choice of $\mathcal{L}$.

$L = \{a^n b^n \mid n \geq 0\}$

$\forall$: chooses some $k > 0$

$\exists$: needs to choose some $xyz \in L$ s.t. $|y| \geq k$

$x = \varepsilon$, $y = a^{2k}$, $z = b^{2k}$

$xyz \in L$, $|y| = 2k > k$.

$\forall$: needs to choose $u, v, w \in \{a, b\}^*$ s.t. $uvw = y$ and $v \neq \varepsilon$.

Suppose $u = a^p$, $v = a^q$, $w = a^r$, s.t. $p+q+r = 2k$, and $q > 0$.

$\exists$: needs to choose $i \geq 0$ s.t. $xuv^iwz \notin L$.

$i = 0$

$\mathcal{L} = \{a^n b^n \mid n \geq 0\}$

$\forall$: chooses some $k > 0$

$\exists$: needs to choose some $xyz \in \mathcal{L}$ s.t. $|y| \geq k$

$x = a^l$, $y = a^k$, $z = a^{n-l-k} b^n$ where $n > k + l$.

$\forall$: needs to choose $u, v, w \in \{a, b\}^*$ s.t. $uvw = y$ and $v \neq \varepsilon$,

Suppose $u = a^p$, $v = a^q$, $w = a^r$, s.t. $p + q + r = k$, and $q > 0$.

$\exists$: needs to choose $i \geq 0$ s.t. $xuv^iwz \notin \mathcal{L}$.