3. Decidability and Complexity

The <u>membership problem</u>, also called <u>word problem</u>, for regular languages is

<u>WORDREG</u>:

<u>Given</u>: An NFA A over Σ and $w \in \Sigma^*$.

<u>Question</u>: Does $w \in L(A)$ hold?

- In general, a <u>decision problem</u> is a set $P \subseteq S$.

Here,

$$\text{WORDREG} \subseteq \text{NFAs} \times \Sigma^*$$

with $(A, w) \in \text{WORDREG}$ iff $w \in L(A)$.

- The problem (the set) P is <u>decidable</u>, if the <u>characteristic function</u>

$$\chi_P : S \to \{0, 1\}$$

is computable.

Recall that for every set $P \subseteq S$, we have (this is the definition)

$$\chi_P(x) = 1 \quad \text{iff} \quad x \in P.$$

More on computability in the corresponding part of the lecture.

<u>Theorem</u>:

WORDREG can be solved in $O(|w| \cdot |Q|^2)$ time.

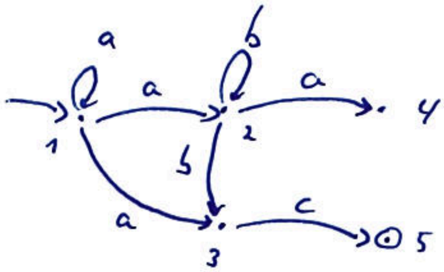
<u>Proof</u>:

<u>Idea</u>: Do a powerset construction <u>along the given word</u>.
Return "yes", if a final state is in the last set of states.

Complexity: We have $|w|$ steps.
In each step, we have to check for at most $|Q|$ states whether there is a transition to any of the $|Q|$ states. $\square$

Example:

$w = abc$ $\qquad$ $\mathcal{A}$:



$\{1\} \xrightarrow{a} \{1,2,3\} \xrightarrow{b} \{2,3\} \xrightarrow{c} \{5\}$.

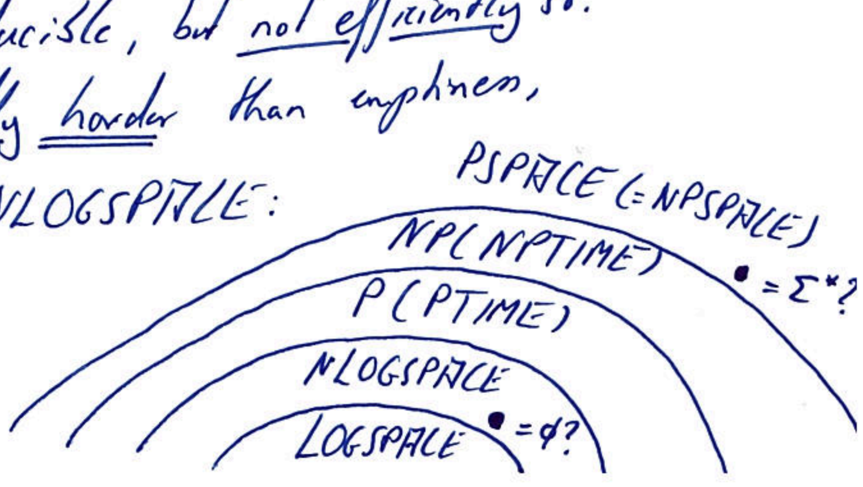
Return "yes".

There are more problems of importance.

EMPTYREG:
Given: $\mathcal{A}$ an NFA.
Question: $L(\mathcal{A}) = \emptyset$?

UNIVERSALITYREG:
Given: $\mathcal{A}$ an NFA over Σ.
Question: $L(\mathcal{A}) = \Sigma^*$?

Remark:
The problems are interreducible, but not efficiently so.
Universality is substantially harder than emptiness,
namely PSPACE vs. NLOGSPACE:
The two classes are known to be different.



Theorem:

EMPTYREG can be solved in $O(|\rightarrow|)$.

Idea:

- Compute the ascending chain

$$R_0 \subseteq R_1 \subseteq R_2 \subseteq \dots$$

of sets of those states that are reachable in $\leq i$ steps.

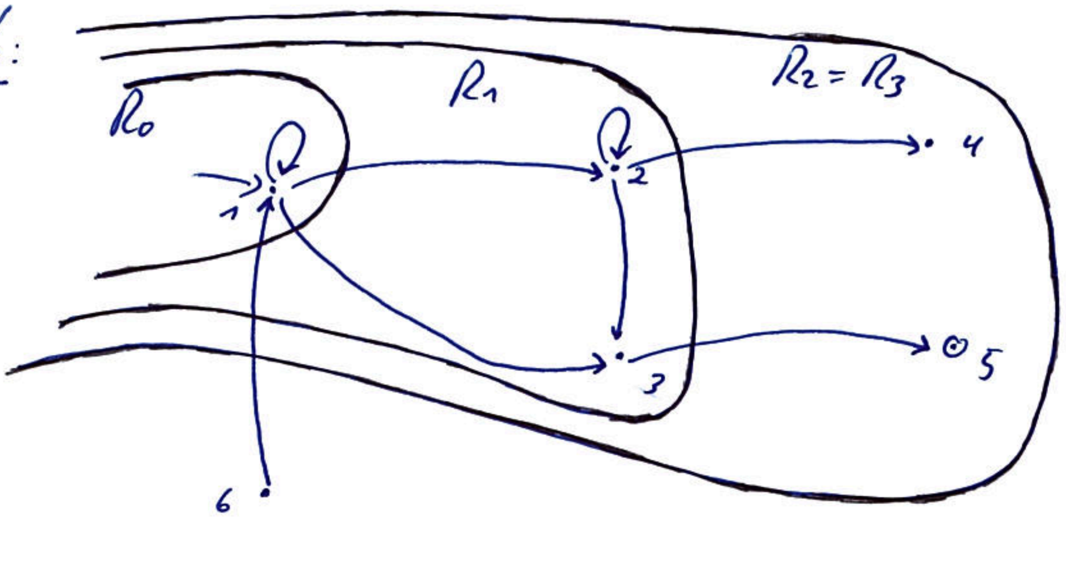
- The computation stops when $R_k = R_{k+1}$.

In this case, the fixed point has been reached:

$$\bigcup_{i \in \mathbb{N}} R_i = R_0 \cup R_1 \cup \dots \cup R_k \cup R_k \cup \dots = R_k.$$

More on fixed points in the semantics part of the lecture.

Example:



Note that the alphabet does not matter.

Proof: Let $A = (\Sigma, Q, q_0, \rightarrow, Q_F)$.

Define $R_0 := \{q_0\}$

$$R_{i+1} := R_i \cup \{q' \in Q \mid \exists q \in R_i\ \exists a \in \Sigma : q \xrightarrow{a} q'\}.$$

Consider $R_k = R_{k+1}$. If $R_k \cap Q_F \neq \emptyset$, return $L(A) \neq \emptyset$, otherwise return $L(A) = \emptyset$.

Complexity:

↳ The fixed point is reached after at most $|Q|$ steps.

This gives complexity $O(|Q| \cdot |{\to}|)$.

↳ It is sufficient to consider every transition $q \xrightarrow{a} q'$ at most once.

This gives $O(|{\to}|)$ as we claimed. $\square$

Further decision problems on NFAs:

INTERSECTION REG: Given NFAs A, B, is $L(A) \cap L(B) \neq \emptyset$?

EQUIVALENCE REG: Given NFAs A, B, is $L(A) = L(B)$?

INCLUSION REG: Given NFAs A, B, is $L(A) \subseteq L(B)$?