Properties of regular languages

Peter Mayr

Computability Theory, September 6, 2023

A necessary condition for regularity

Question

Is every language L over Σ regular? How to show it is not?

No, uncountably many subsets of Σ^* (languages).
but only countable many regular expression.

Pumping Lemma

For any regular language L there exists $n \in \mathbb{N}, n \neq 0$ (the **pumping length** of L) such that $\forall w \in L, |w| \geq n, \exists x, y, z \in \Sigma^*$ such that

- $w = xyz$
- $y \neq \epsilon$
- $|xy| \leq n$
- $\forall k \in \mathbb{N}\colon xy^k z \in L$.

Example

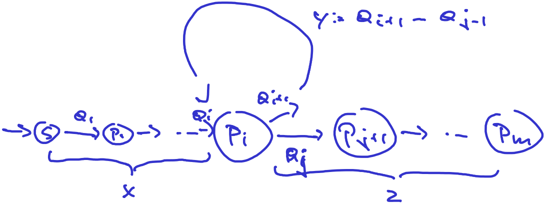
$\{0^k 1^k \ : \ k \in \mathbb{N}\}$ is not regular since it does not have any pumping length.

Proof.

Let $L = L(M)$ for a DFA M with n states.
Let $w = a_1 \dots a_m \in L$ with $a_i \in \Sigma$ and $m \geq n$. Define

$$p_i := \delta^*(s, a_1 \dots a_i) \text{ for } i \leq m.$$

By the pigeonhole principle $\exists 0 \leq i < j \leq n$: $p_i = p_j$.
Consider the path labelled by w:



$w = xyz$

Then $\delta^*(p_i, y) = p_i$ and $xy^k z \in L$ for all $k \in \mathbb{N}$. $\square$

Myhill-Nerode Theory

For $L \subseteq \Sigma^*$ define an equivalence relation R_L on Σ^* by

$$x\, R_L\, y \text{ if } \forall w \in \Sigma^* \colon (xw, yw \in L) \text{ or } (xw, yw \in \Sigma^* \setminus L).$$

Idea: If L is regular and x, y are not related, then they correspond to different states $\delta^*(s, x) \neq \delta^*(s, y)$.
Note: The action of the semigroup Σ^* on Σ / R_L is welldefined.

Theorem (Myhill, Nerode 1958)

L is regular iff Σ^* / R_L is finite.

Example

$L = \{0^k 1^k \ : \ k \in \mathbb{N}\}$ is not regular since 0^k for $k \in \mathbb{N}$ are in pairwise distinct R_L-classes.

Proof.

$\Rightarrow$: Let $L = L(M)$ for a DFA M with states $\{1, \dots, n\}$ and start state 1.

For $i \leq n$, define

$$S_i := \{w \in \Sigma^* \ : \ \delta^*(1, w) = i\}.$$

Then $S_1, \dots, S_n$ refine $\Sigma^*/\mathbb{R}_L$ and $|\Sigma^*/\mathbb{R}_L| \leq n$.

$\Leftarrow$: Define a DFA M_L with

- $\Sigma^*/R_L = \{S_1, \dots, S_n\} =: Q$ (states)
- $\delta(w/R_L, a) := wa/R_L$ (transition function, welldefined!)
- $s := \epsilon/R_L$ (start state)
- $F := \{w/R_L \ : \ w \in L\}$ (final states)

Then $L(M_L) = L$. $\square$

Corollary

M_L above is the unique minimal DFA that accepts L.

Summary on automata and regular languages

- Converting NFA to DFA increases the number of states exponentially (in the worst case).
- Converting DFA to regular expresssions (or conversely) is exponential in the number of states (the length of the expression).
- **Membership in $L(M)$:** Easy, check $w \in L(M)$ by running M with input w (takes $|w|$ steps).
- **Emptiness of $L(M)$:** Easy, check whether some final state is reachable from the start state (cf. graph reachability, takes n^2 steps).
- **Universality of $L(M)$:** Every $w \in \Sigma^*$ is accepted by DFA M iff ...