# Rozhodnutelné problémy pro třídu regulárních jazyků

V předchozím dokumentu jsme se začali regulárními jazyky zabývat obecně a ukázali jsme si jejich uzavřenost na různé operace, se kterými jsme se již setkali. Nyní nás budou zajímat ještě obecnější problémy a otázka, zda se dají algoritmicky řešit. Popíšeme si je pomocí dvou konečných automatů $M, M'$ nad abecedou $\Sigma$:

1. **Problém ekvivalence** $L(M) = L(M')$;
2. **Problém inkluze** $L(M) \subseteq L(M')$;
3. **Problém příslušnosti** slova $w \in \Sigma^*$ do jazyka $L(M)$ (tj. pro zadané slovo $w$ platí $w \in L(M)$?);
4. **Problém prázdnosti** jazyka $L(M)$, tj. zda $L(M) = \emptyset$;
5. **Problém univerzality** jazyka $L(M)$, tj. zda $L(M) = \Sigma^*$;
6. **Problém konečnosti** jazyka $L(M)$.

Naznačíme si, jak lze uvedené problémy algoritmicky řešit.

## Problém ekvivalence

Existuje dvojí cesta, přičemž jedna už je vám známá. V případě, že máme dva konečné automaty $M_1, M_2$ a chceme zjistit, zda generují tentýž jazyk $L$, pak můžeme provést minimalizaci obou automatů a následně je „kanonizovat“. Jsou-li oba redukované automaty stejné až na pojmenování stavů, pak generují tentýž jazyk.

## Poznámka 1.

Konečný automat je **kanonickém tvaru**, právě když jeho stavy jsou uspořádány tak, že počáteční stav je první v pořadí, a konečné stavy jsou v uspořádání na samotném konci.

Další, a patrně méně časově náročnou možností, je aplikace následujícího tvrzení: pro libovolné jazyky $L_1, L_2$ nad abecedou $\Sigma$ platí, že:

$$(L_1 = L_2) \quad \Longleftrightarrow \quad (L_1 \cap \text{co-}L_2) \cup (\text{co-}L_1 \cap L_2) = \emptyset$$

Na této rovnosti je založen následující algoritmus:

## Algoritmus (test rovnosti jazyků dvou automatů).

**Vstup:** dva konečné automaty $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1),\ M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.
**Výstup:** **ANO**, pokud $L(M_1) = L(M_2)$;
**NE** v opačném případě.
$i := 0$
$R_1 = \{(q_1, q_2)\}$
**repeat**
$R_{i+1} = R_i \cup \{(p, q) \mid \delta_1(r, a) = p, \delta_2(s, a) = q,\ (r, s) \in R_i, a \in \Sigma\}$
if $R_{i+1}$ obsahuje dvojici $(k, l)$, kde $k \in F_1 \Leftrightarrow l \notin F_2$ **then return NE**
$i := i + 1$
**until** $(R_i = R_{i-1})$
**return ANO**

## Poznámka 2.

Algoritmus spočívá v konstrukci množiny uspořádaných dvojic stavů, do kterých se dostaneme při výpočtu obou automatů na nějakém slově. Začínáme tak, že do množiny $R$ přidáme dvojici složenou z počátečných stavů obou automatů, tedy $R = \{q_1, q_2\}$. Následně zkoumáme, kam se dostaneme z obou stavů pod libovolným symbolem $a \in \Sigma$, a výsledné dvojice stavů vložíme do $R$.

Pro každou $(r, s) \in R$ ověřujeme, zda oba stavy $r, s$ jsou současně buď koncové, nebo nekoncové. Když např. $r \in F_1 \wedge s \notin F_2$, algoritmus končí s výsledkem **NE**, tj. $L(M_1) \neq L(M_2)$. Stejně tak v opačném případě, tj. $r \notin F_1 \wedge s \in F_2$.

Dokud můžeme, přidáváme do množiny $R$ stále nové dvojice a zkoušíme, kam se z nich dostaneme v obou automatech. Algoritmus končí, jakmile se nám nepodaří vložit novou dvojici stavů, která v $R$ předtím nebyla. V takovém případě je výsledek **ANO**, tj. $L(M_1) = L(M_2)$.

## Příklad 1.

Jsou dány dva konečné automaty

$$\begin{aligned} M_1 &= (\{p_0, p_1, p_2, p_3\}, \{a, b\}, \delta_1, p_0, \{p_3\}), \\ M_2 &= (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta_2, q_0, \{q_1\}), \end{aligned}$$

kde přechodové funkce $\delta_1, \delta_2$ jsou následující:

| | $\delta_1$ | $a$ | $b$ |
|---|---|---|---|
| $\rightarrow$ | $p_0$ | $p_1$ | $p_0$ |
| | $p_1$ | $p_1$ | $p_2$ |
| | $p_2$ | $p_3$ | $p_0$ |
| $\leftarrow$ | $p_3$ | $p_3$ | $p_3$ |

| | $\delta_2$ | $a$ | $b$ |
|---|---|---|---|
| $\rightarrow$ | $q_0$ | $q_2$ | $q_0$ |
| $\leftarrow$ | $q_1$ | $q_0$ | $q_0$ |
| | $q_2$ | $q_2$ | $q_3$ |
| | $q_3$ | $q_1$ | $q_0$ |

Rozhodněte, zda $L(M_1) = L(M_2)$.

***Řešení.*** Položme $F_1 = \{p_3\}, F_2 = \{q_1\}$. Do množiny $R$ vložíme dvojici $(p_0, q_0)$ a postupně ji naplňujeme:

1. Pro $a$ platí: $\delta_1(p_0, a) = p_1,\ \delta_2(q_0, a) = q_2$,
   pro $b$ platí: $\delta_1(p_0, b) = p_0,\ \delta_2(q_0, a) = q_0$,
   tedy $R = \{(p_0, q_0), (p_1, q_2)\}$. V algoritmu pokračujeme, protože $p_1 \notin F_1, q_2 \notin F_2$.

2. Do $R$ jsme přidali novou dvojici $(p_1, q_2)$, zkontrolujeme její přechody.
   Symbol $a$: $\delta_1(p_1, a) = p_1,\ \delta_2(q_2, a) = q_2$,
   symbol $b$: $\delta_1(p_1, a) = p_2,\ \delta_2(q_2, b) = q_3$,
   což dává $R = \{(p_0, q_0), (p_1, q_2), (p_2, q_3)\}$. U nově přidané dvojice platí $p_2 \notin F_1, q_3 \notin F_2$, takže pokračujeme v hledání prvků množiny $R$ dále.

3. Zapíšeme přechody pro dvojici $(p_2, q_3)$:
   symbol $a$: $\delta_1(p_2, a) = p_3,\ \delta_2(q_3, a) = q_1$,
   symbol $b$: $\delta_1(p_2, b) = p_0,\ \delta_2(q_3, b) = q_0$,
   čili do $R$ opět přidáváme novou dvojici $(p_3, q_1)$. Pro ni však platí, že $p_3 \in F_1, q_1 \in F_2$. Pokračujeme dále, víme, že

$$R = \{(p_0, q_0), (p_1, q_2), (p_2, q_3), (p_3, q_1)\}$$

4. Napíšeme si přechody pro dvojici $(p_3, q_1)$:
   symbol $a$: $\delta_1(p_3, a) = p_3,\ \delta_2(q_1, a) = q_0$,
   symbol $b$: $\delta_1(p_3, b) = p_3,\ \delta_2(q_1, b) = q_0$.
   V obou případech se jedná o tutéž dvojici $(p_3, q_0)$, přičemž $p_3 \in F_1$, kdežto $q_0 \notin F_2$.

Algoritmus končí s výsledkem **NE** a platí $L(M_1) \neq L(M_2)$.

## Příklad 2.

Jsou dány dva konečné automaty

$$\begin{aligned} M_1 &= (\{p_0, p_1, p_2, p_3, p_4\}, \{a, b\}, \delta_1, p_0, \{p_4\}), \\ M_2 &= (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta_2, q_0, \{q_3\}), \end{aligned}$$

kde přechodové funkce $\delta_1, \delta_2$ jsou následující:

| | $\delta_1$ | $a$ | $b$ |
|---|---|---|---|
| $\rightarrow$ | $p_0$ | $p_1$ | $p_2$ |
| | $p_1$ | $p_2$ | $p_1$ |
| | $p_2$ | $p_3$ | $p_2$ |
| | $p_3$ | $p_1$ | $p_4$ |
| $\leftarrow$ | $p_4$ | $p_4$ | $p_4$ |

| | $\delta_2$ | $a$ | $b$ |
|---|---|---|---|
| $\rightarrow$ | $q_0$ | $q_1$ | $q_0$ |
| | $q_1$ | $q_2$ | $q_1$ |
| | $q_2$ | $q_0$ | $q_3$ |
| $\leftarrow$ | $q_3$ | $q_3$ | $q_3$ |

Rozhodněte, zda $L(M_1) = L(M_2)$.

## Problém příslušnosti

Buď $M = (Q, \Sigma, \delta, q_0, F)$ konečný automat pro jazyk $L$, $w \in \Sigma^*$ slovo. Pokud $M$ dokážeme převést na deterministický konečný automat s totální přechodovou funkcí (což umíme), je příslušnost do jazyka otázkou výpočtu automatu, který buď skončí v koncovém stavu ($w \in L$) nebo ne ($w \notin L$).

## Problém prázdnosti jazyka

V kapitole o minimalizaci jsme si uváděli algoritmus na odstranění nedosažitelných stavů. Jazyk přijímaný automatem $M = (Q, \Sigma, \delta, q_0, F)$ je prázdný, pokud aplikací tohoto algoritmu zjistíme, že ani jeden z koncových stavů $q \in F$ není dosažitelný.

## Problém univerzality jazyka

Je algoritmicky řešitelný, pokud si uvědomíme platnost následující ekvivalence

$$L(M) = \emptyset \quad \Longleftrightarrow \quad \text{co-}L(M) = \Sigma^*$$

pro libovolný konečný automat $M$ nad abecedou $\Sigma$. Ptáme-li se, zda jazyk $L(M) = \Sigma^*$, stačí sestrojit automat $M'$ pro jazyk co-$L(M)$ a pomocí algoritmu pro odstranění nedosažitelných stavů zjistit, zda $M'$ přijímá prázdný jazyk.

## Problém konečnosti

Problém, zda jazyk $L(M)$ je konečný či nekonečný, je rozhodnutelný díky platnosti následující věty:

## Věta 1 (o nekonečném jazyku).

Buď $M = (Q, \Sigma, \delta, q_0, F)$ konečný automat, kde $|Q| = n,\ n \in \mathbb{N}$. Jazyk $L(M)$ je nekonečný, právě když existuje slovo $w \in L(M)$ takové, že $n \leq |w| < 2n$.

## Důkaz.

Uvedená věta je tvaru ekvivalence $P \Leftrightarrow Q$, kde $P$ je tvrzení, že jazyk $L(M)$ je nekonečný. Symbol $Q$ označuje existenci slova $w \in L(M)$, pro které platí $n \leq |w| < 2n$. Budeme tedy ukazovat oba směry, a to $P \Rightarrow Q$ i $P \Leftarrow Q$:

$\Rightarrow$: Předpokládejme, že jazyk $L$ je nekonečný. Určitě existuje slovo $w \in \Sigma^*$, které je delší než $n$ ($|w| \geq n$).

Pokud $|w| < 2n$, je důkaz hotov. V opačném případě ($|w| \geq n$) použijeme lemma o vkládání: z platnosti $w \in L(M),\ |w| \geq n$ vyplývá existence rozdělení $w = xyz$, kde $|xy| \leq n,\ |y| \geq 1,\ xy^iz \in L(M)$. Je patrné, že $|xy| \leq n \Rightarrow |y| \leq n$. Protože $|xyz| \geq 2n$, zbývá na části $x, z$ minimálně $n$ symbolů, tj. $|xz| \geq n$.

Uvažujme slovo $w = xz$. Pokud $|xz| < 2n$, nalezli jsme slovo $w$ požadované vlastnosti $n \leq |w| < 2n$. V opačném případě opakujeme postup z předchozího odstavce.

Celý postup opakujeme tak dlouho, dokud se nám slovo $w,\ n \leq |w| < 2n$ nepodaří nalézt. Určitě uspějeme.

$\Leftarrow$: Předpokládejme, že máme slovo $w \in L(M)$ takové, že $n \leq |w| < 2n$. Protože $|w| \geq n$ můžeme uplatnit Lemma o vkládání, tj. rozdělit slovo $w$ na 3 části $w = xyz$, kde $|y| \geq 1,\ |xy| \leq n$ a $xy^iz \in L(M)$, kde $i \in \mathbb{N}_0$. Jazyk $L$ je tedy nekonečný, protože množina $\{xy^iz, i \in \mathbb{N}_0\}$ je nekonečná.

## Poznámka 3.

Problém konečnosti jazyka je tedy algoritmicky řešitelný. Jestliže máme konečný automat $M$ o $n$ stavech, pak je možné ověření existence slova $w \in L(M)$, jehož délka je zdola omezena $n$ a shora $2n$. Při konečnosti abecedy $\Sigma$ musíme projít konečně mnoho slov $w \in \Sigma^*$ délky minimálně $n$ a maximálně $2n$.

## Příklad 3.

Ověřte, zda jazyk $L(M)$ přijímaný konečným automatem $M$ je nekonečný.
$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_0, \{q_3, q_4\})$, kde $\delta$ je dána tabulkou:

| | $\delta$ | $a$ | $b$ |
|---|---|---|---|
| $\rightarrow$ | $p_0$ | $q_1$ | $q_2$ |
| | $q_1$ | $q_3$ | $q_1$ |
| | $q_2$ | $q_2$ | $q_4$ |
| $\leftarrow$ | $q_3$ | $q_3$ | $q_1$ |
| $\leftarrow$ | $q_4$ | $q_2$ | $q_4$ |

***Řešení.*** Automat $M$ má 5 stavů, stačí tedy nalézt slovo $w \in L(M)$, jehož délka je zdola omezena číslem 5, shora číslem 10. Např. $abbbba \in L(M),\ |abbbba| = 6$. Jazyk $L(M)$ je nekonečný.