2. Vlastnosti regulárních jazyků a nedeterminismus

Formální jazyky a automaty

Jiří Balun

Obsah

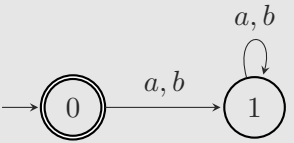
1. Základní vlastnosti regulárních jazyků
   - Doplněk
   - Součinový DFA
   - Průnik a sjednocení
   - Nedosažitelné stavy
   - Test na neprázdnost
   - Rovnost
   - Inkluze
2. Nedeterministický konečný automat (NFA)
   - Popis NFA
   - Jazyk NFA
   - NFA s více počátečními stavy

Příklad: jazyk obsahující jen prázdný řetězec

- nejprve si představíme tři jednoduché jazyky, na nichž budeme vlastnosti demonstrovat
- pro účely této přednášky si jednotlivé jazyky a automaty pojmenujeme

Příklad

Mějmě jazyk $L_\varepsilon = \{\varepsilon\}$ obsahující pouze prázdný řetězec.

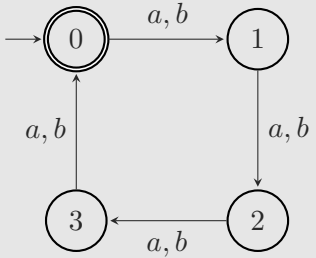


- DFA který jej rozpoznává označíme $A_\varepsilon = (Q_\varepsilon, \{a, b\}, \delta_\varepsilon, 0, \{0\})$
- A_ε zamítne pokud vstupní řetězec obsahuje aspoň jeden znak

Příklad: počítání délky slova

Příklad

Mějmě jazyk $L_{mod} = \{w \in \{a,b\}^* \mid \text{platí, že } |w| \equiv 0 \,(\text{mod } 4)\}$.

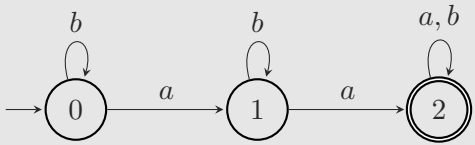


- DFA který jej rozpoznává označíme $A_{mod} = (Q_{mod}, \{a,b\}, \delta_{mod}, 0, \{0\})$
- každý stav (jeho pojmenování) odpovídá aktuální délce slova modulo 4
- přidáním/odebráním stavů lze zobecnit pro jazyky slov délky pro obecné $n \in \mathbb{N}$

Příklad: počítání výskytů znaku

Příklad

Mějmě jazyk $L_{\#} = \{w \in \{a,b\}^* \mid w$ obsahuje alespoň 2 znaky a, tj. $\#_a(w) = 2\}$.



- DFA který jej rozpoznává označíme $A_{\#} = (Q_{\#}, \{a,b\}, \delta_{\#}, 0, \{2\})$
- každý stav (jeho pojmenování) odpovídá aktuálnímu počtu přečtených symbolů a

Doplněk regulárního jazyka

Věta

Nechť $L \subseteq \Sigma^*$ je regulární jazyk, pak jazyk $\bar{L} = \Sigma^* \setminus L$, který je jeho doplňkem vzhledem k univerzu Σ^*, je také regulární.

Důkaz

Idea: V automatu, který rozpoznává L prohodíme koncové a nekoncové stavy.

Jazyk L je regulární, proto existuje DFA $A = (Q, \Sigma, \delta, q_0, F)$ takový, že $L(A) = L$.
Sestrojíme nový DFA $\bar{A} = (Q, \Sigma, \delta, q_0, Q \setminus F)$ a ukážeme, že $\bar{L} = L(\bar{A}) = \Sigma^* \setminus L(A)$:

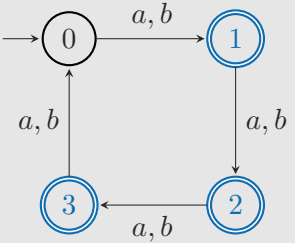
- dle definice $w \in L(\bar{A})$ právě když $\delta(q_0, w) \in Q \setminus F$,
- to je ekvivaletní s $\delta(q_0, w) \notin F$, a tedy platí $w \notin L(A)$. □

- **Důsledek:** třída regulárních jazyků je uzavřená na doplněk
- všimněte si, že důkaz nefunguje pokud by DFA A měl neúplnou přechodovou funkci

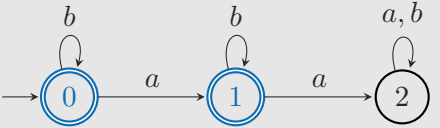
Příklad: doplněk

Příklad

DFA rozpoznávající jazyky $\bar{L}_{mod}$ a $\bar{L}_{\#}$:



a) DFA $\bar{A}_{mod}$



b) DFA $\bar{A}_{\#}$

a) $\bar{A}_{mod} = (Q_{mod}, \{a, b\}, \delta_{mod}, 0, \{1, 2, 3\})$

- rozpoznává jazyk $\bar{L}_{mod} = \{w \in \{a, b\}^* \mid \text{platí, že } |w| \not\equiv 0 \,(\text{mod } 4)\}$

b) $\bar{A}_{\#} = (Q_{\#}, \{a, b\}, \delta_{\#}, 0, \{0, 1\})$

- rozpoznává jazyk $\bar{L}_{\#} = \{w \in \{a, b\}^* \mid w \text{ obsahuje nanejvýš 1 znak } a, \text{ tj. } \#_a(w) \leq 1\}$

Součinový DFA

Intuice: součinový automat simuluje běh více automatů najednou.

Definice

Pro libovolné dva DFA $A_1 = (Q_1, \Sigma, \delta_1, q_{0,1}, F_1)$ a $A_2 = (Q_2, \Sigma, \delta_2, q_{0,2}, F_2)$ nad abecedou Σ definujeme **součinový DFA** jako $A_1 \times A_2 = (Q_1 \times Q_2, \Sigma, \delta, \langle q_{0,1}, q_{0,2} \rangle, F)$, kde:

- množina stavů obsahuje všechny **dvojice stavů** $\langle q_1, q_2 \rangle$, kde $q_1 \in Q_1$ a $q_2 \in Q_2$,
- abeceda Σ zůstává stejná (A_1 a A_2 musí být nad stejnou abecedou),
- $\delta(\langle q_1, q_2 \rangle, a) = \langle \delta_1(q_1, a), \delta_2(q_2, a) \rangle$ pro všechna $a \in \Sigma$,
- počáteční stav $\langle q_{0,1}, q_{0,2} \rangle$ je dvojice počátečních stavů z A_1 a A_2,
- množinu koncových stavů F zvolíme podle účelu $A_1 \times A_2$ (viz. dále).

- konstrukci lze analogicky rozšířit pro libovolný konečný součin automatů $A_1 \times \ldots \times A_n$
- pokud A_1 a A_2 jsou nad různými abecedami $\Sigma_1 \neq \Sigma_2$, lze je "rozšířit":
  - oběma nastavíme $\Sigma_1 \cup \Sigma_2$ a poté zúplníme jejich přechodové funkce

Průnik regulárních jazyků

Intuice: použijeme součinový automat, který přijme pokud oba simulované automaty přijmou, tj. zvolíme koncové stavy $F = \{\langle q_1, q_2 \rangle \mid q_1 \in F_1 \wedge q_2 \in F_2\}$.

Věta

Pokud L_1 a L_2 jsou regulární jazyky, pak $L_1 \cap L_2$ je také regulární jazyk.

Důkaz

K jazykům L_1 a L_2 existují automaty A_1 a A_2, takové že:

1. $A_1 = (Q_1, \Sigma, \delta_1, q_{0,1}, F_1)$ a $L(A_1) = L_1$,
2. $A_2 = (Q_2, \Sigma, \delta_2, q_{0,2}, F_2)$ a $L(A_2) = L_2$.

Zkonstruujeme $A_1 \times A_2$ s množinou koncových stavů $F = \{\langle q_1, q_2 \rangle \mid q_1 \in F_1 \wedge q_2 \in F_2\}$.

Potom platí: $w \in L(A_1 \times A_2)$ právě když $\delta(\langle q_{0,1}, q_{0,2} \rangle, w) \in F$, což je ekvivalentní s $\delta_1(q_{0,1}, w) \in F_1$ a zároveň $\delta_2(q_{0,2}, w) \in F_2$, a proto $w \in L_1 \cap L_2$. □

- **Důsledek:** třída regulárních jazyků je uzavřená na průnik

Sjednocení regulárních jazyků

Intuice: použijeme součinový automat, který přijme pokud alespoň jeden simulovaný automat přijme, tj. zvolíme koncové stavy $F = \{\langle q_1, q_2 \rangle \mid q_1 \in F_1 \vee q_2 \in F_2\}$.

Věta

Pokud L_1 a L_2 jsou regulární jazyky, pak $L_1 \cup L_2$ je také regulární jazyk.

Důkaz

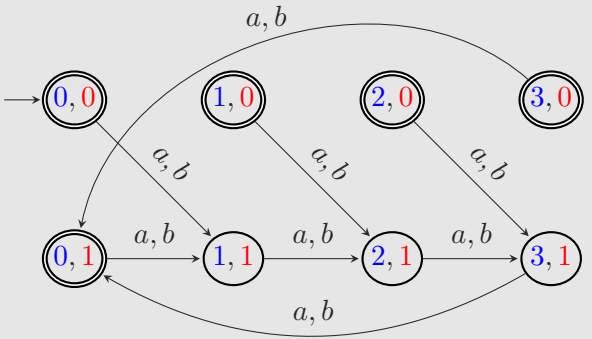
Analogicky jako u průniku. □

- **Důsledek:** třída regulárních jazyků je uzavřená na sjednocení

Příklad: sjednocení regulárních jazyků

Příklad

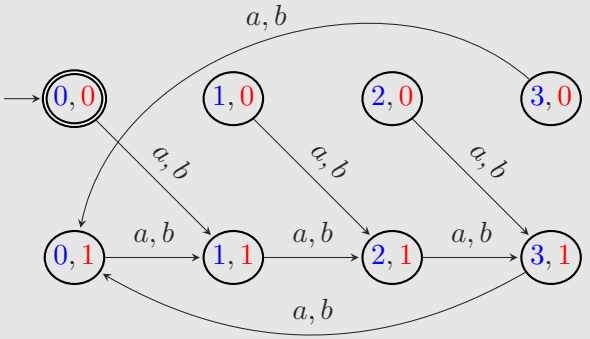
Součinový DFA $A_{mod} \times A_{\varepsilon}$ s množinou koncových stavů $F = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 2,0\rangle, \langle 3,0\rangle, \langle 0,1\rangle\}$ rozpoznává jazyk $L(A_{mod}) \cup L(A_{\varepsilon})$:



Příklad: průnik regulárních jazyků

Příklad

Součinový DFA $A_{mod} \times A_{\varepsilon}$ s množinou koncových stavů $F = \{\langle 0, 0 \rangle\}$ rozpoznává jazyk $L(A_{mod}) \cap L(A_{\varepsilon}) = L_{\varepsilon}$:



Nedosažitelné stavy

Intuice: stav q je dosažitelný pokud do něj v grafu automatu vede cesta z počátečního stavu, tj. $\delta(q_0, w) = q$ pro nějaké $w \in \Sigma^*$. V opačném případě je q nedosažitelný.

Věta

Ke každému DFA $A = (Q, \Sigma, \delta, q_0, F)$ existuje DFA A^{acc} takový, že všechny stavy A^{acc} jsou dosažitelné a zároveň $L(A) = L(A^{acc})$.

Důkaz

Idea: Zkonstruujeme automat A^{acc} tak, že "zmenšíme" A.

1. Množinu dosažitelných stavů v A definujeme jako $S = \{q \mid \exists w \in \Sigma^* \colon \delta(q_0, w) = q\}$,
2. poté definujeme $A^{acc} = (S, \Sigma, \delta^{acc}, q_0, F \cap S)$, kde $\delta^{acc} = \delta \cap (S \times \Sigma \times S)$,
3. evidentně platí $L(A^{acc}) \subseteq L(A)$, jelikož A^{acc} vznikl jen odebráním stavů,
4. $L(A) \subseteq L(A^{acc})$ platí protože pro každé $w \in L(A)$ je $\delta(q_0, w) = q_f \in F$, kde q_f je dosažitelný stav, a tedy $q_f \in S$ a $\delta^{acc}(q_0, w) \in F \cap S$. □

Test na neprázdnost jazyka

Intuice: jazyk libovolného automatu je neprázdný právě když je dosažitelný aspoň jeden jeho koncový stav.

Věta

Mějme DFA $A = (Q, \Sigma, \delta, q_0, F)$, pak $L(A) \neq \emptyset$ právě když existuje takový $w \in \Sigma^*$, že platí $\delta(q_0, w) \in F$.

Důkaz

Stačí vypočítat A^{acc} a ověřit, zda je jeho množina koncových stavů F^{acc} prázdná. Pokud tedy $F^{acc} \neq \emptyset$, pak libovolný $q_f \in F^{acc}$ je dosažitelný a dle definice existuje $w \in \Sigma^*$ takový, že $\delta(q_0, w) \in F^{acc}$. V opačném případě $F^{acc} = \emptyset$, a tedy $L(A^{acc}) = \emptyset = L(A)$. □

- tento problém použijeme k rozhodování inkluze a rovnosti regulárních jazyků
- A^{acc} lze vypočítat pomocí klasických algoritmů na průchod grafu v polynomiálním čase

Rozhodování rovnosti regulárních jazyků

Intuice: rozhodneme neprázdnost jazyka součinového automatu, který přijmá v případě, že pouze jeden simulovaný automat přijme (nalezne protipříklad), tj. zvolíme koncové stavy:

$$F = \{\langle q_1, q_2 \rangle \mid (q_1 \in F_1 \wedge q_2 \notin F_2) \vee (q_1 \notin F_1 \wedge q_2 \in F_2)\}$$

Věta

Problém rovnosti $L_1 = L_2$ dvou regulárních jazyků L_1 a L_2 je rozhodnutelný.

Důkaz

K jazykům L_1 a L_2 existují automaty A_1 a A_2, takové že:

- $A_i = (Q_i, \Sigma, \delta_i, q_{0,i}, F_i)$ a $L(A_i) = L_i$ pro $i \in \{1, 2\}$.

Zkonstruujeme $A_1 \times A_2$ s množinou koncových stavů F pro který platí, že $L_1 = L_2$ právě když $L(A_1 \times A_2) = \emptyset$.

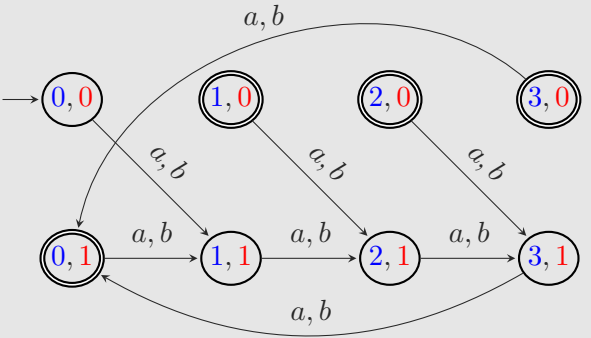
Platí: $w \in L(A_1 \times A_2) \neq \emptyset$ právě když $\delta(\langle q_{0,1}, q_{0,2} \rangle, w) \in F$, což je ekvivalentní s:

1. $\delta_1(q_{0,1}, w) \in F_1$ a zároveň $\delta_2(q_{0,2}, w) \notin F_2$, a proto $w \in L_1 \cap \bar{L}_2$, nebo
2. $\delta_1(q_{0,1}, w) \notin F_1$ a zároveň $\delta_2(q_{0,2}, w) \in F_2$, a proto $w \in \bar{L}_1 \cap L_2$. □

Příklad: rovnost regulárních jazyků

Příklad

Součinový DFA $A_{mod} \times A_{\varepsilon}$ s množinou koncových stavů $F = \{\langle 0,1\rangle, \langle 1,0\rangle, \langle 2,0\rangle, \langle 3,0\rangle\}$ rozpoznává jazyk $(L_{mod} \cap \bar{L}_{\varepsilon}) \cup (\bar{L}_{mod} \cap L_{\varepsilon}) \neq \emptyset$, a proto $L_{mod} \neq L_{\varepsilon}$ (protipříklad $w = aaaa$):



Inkluze regulárních jazyků

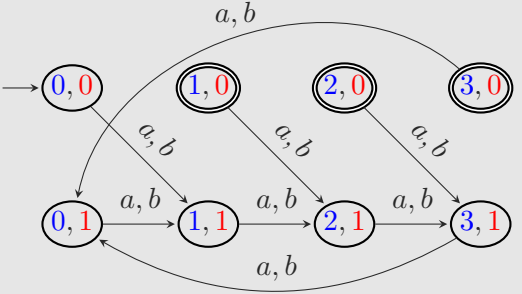
Intuice: analogicky jako u rovnosti, jen zvolíme $F = \{\langle q_1, q_2 \rangle \mid q_1 \in F_1 \wedge q_2 \notin F_2\}$.

Věta

Problém inkluze $L_1 \subseteq L_2$ dvou regulárních jazyků L_1 a L_2 je rozhodnutelný. □

Příklad

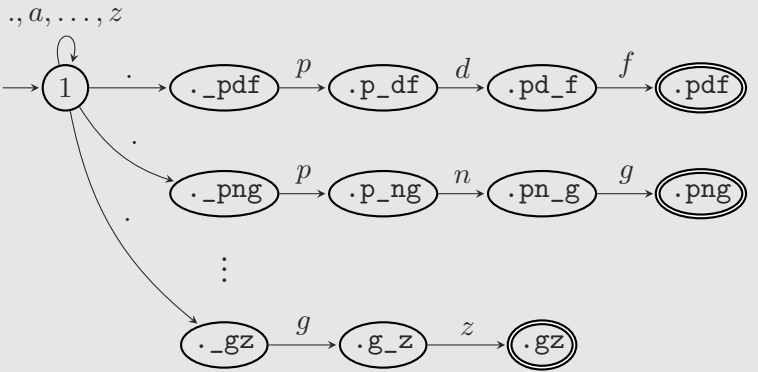
Součinový DFA $A_{mod} \times A_{\varepsilon}$ s množinou koncových stavů $F = (Q_{mod} \setminus F_{mod}) \times F_{\varepsilon} = \{\langle 1, 0 \rangle, \langle 2, 0 \rangle, \langle 3, 0 \rangle\}$ rozpoznává jazyk $\bar{L}_{mod} \cap L_{\varepsilon} = \emptyset$, a proto $L_{\varepsilon} \subseteq L_{mod}$:



Motivační příklad pro nedeterminisus

Příklad

Chceme filtrovat soubory učitého typu s odpovádající příponou, jako například `.pdf`, `.png`, `.gz`, `.tar.gz` atd. přičemž i samotné jméno souboru může obsahovat tečku.



Nedeterministický konečný automat

Intuice: rozšíříme definici DFA o možnost být ve více stavech na jednou.

Definice

Nedeterministický konečný automat (zkráceně NFA z *nondeterministic finite automaton*) je reprezentován uspořádanou pěticí $A = (Q, \Sigma, \delta, q_0, F)$, kde:

1. Q je konečná množina stavů
2. Σ je abeceda
3. δ je přechodová funkce ve tvaru $\delta : Q \times \Sigma \to 2^Q$, která stavu a symbolu přiřadí množinu stavů
4. q_0 je počáteční stav (platí $q_0 \in Q$)
5. F je množina koncových/akceptujících stavů (platí $F \subseteq Q$)

- jediná změna oproti definici DFA je v přechodové funkci:
  - přechodová funkce nemusí být uplná, může tedy nastat $\delta(q, a) = \emptyset$ (protože $\emptyset \in 2^Q$)
  - z jednoho stavu může vést více přechodů pod stejným symbolem, například $\delta(q, a) = \{r, s\}$

Rozšířená přechodová funkce

Intuice: vzhledem k tomu, že NFA se může nacházet ve více stavech najednou, musíme umět vypočítat přechod i z libovolné podmnožiny stavů.

Definice

Rozšířenou přechodovou funkci $\hat{\delta}\colon 2^Q \times \Sigma^* \to 2^Q$ definujeme pro dvojici množiny stavů $R \subseteq Q$ a řetězce $s = s_1 \dots s_n \in \Sigma^*$ následovně:

$$\hat{\delta}(R, s) = \begin{cases} R & \text{pro } s = \varepsilon \\ \hat{\delta}\big(\bigcup_{r \in R} \delta(r, s_1), s_2 \dots s_n\big) & \text{pro } s \neq \varepsilon \end{cases}$$

- $\hat{\delta}$ nám dává návod jak vypočítat množinu stavů dosažitelných nějakým řetězcem
- opět nebudeme rozlišovat mezi δ a $\hat{\delta}$, tj. budeme psát rovnou $\delta(q, s_1 s_2 s_3) = \{r, s\}$

Jazyk NFA

Intuice: NFA přijímá řetězec, pokud po jeho přečtení skončí alespoň v jednom koncovém stavu.

Definice

Jazyk NFA $A = (Q, \Sigma, \delta, q_0, F)$ definujeme jako $L(A) = \{s \mid \hat{\delta}(q_0, s) \cap F \neq \emptyset\}$.

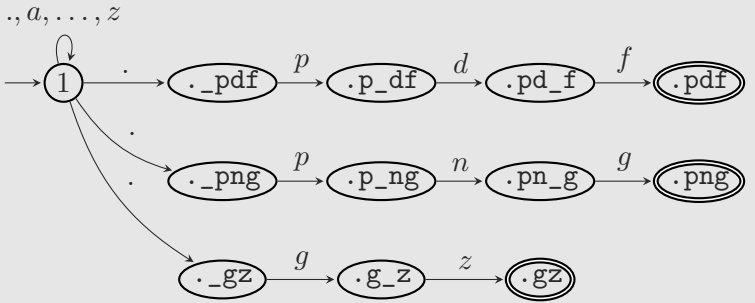
Druhý pohled na nedeterminismus

- NFA se nachází pouze v jednom stavu (jako DFA)
- navíc ale umí navíc "správně" hádat:
  - když má na výběr z více přechodů vždy správně uhádne cestu ke koncovému stavu, pokud taková cesta existuje
- toto chápání nedeterminismu je častější u složitějších modelů (zásobníkový automat, Turingův stroj atd.), které mezi stavy i modifikují svojí paměť

Příklad: výpočet NFA

Příklad

NFA $A_{sfx} = (Q, \{., a, \ldots, z\}, \delta, 1, \{\texttt{.pdf}, \texttt{.png}, \texttt{.gz}\})$, který rozpoznává jazyk přípon souborů $L = \{w \mid w$ má příponu `.pdf`, `.png`, nebo `.gz`$\}$.



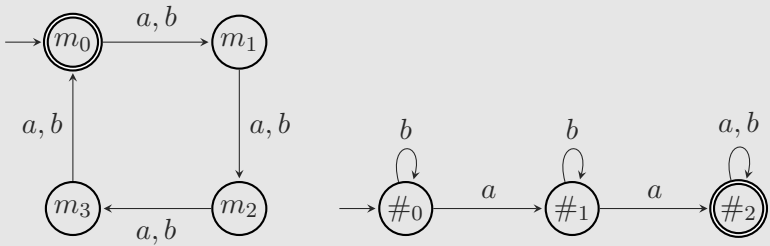
- A_{sfx} přijímá řetězec $.gz$, protože $\hat{\delta}(1, .gz) \cap F \neq \emptyset$:
  $\hat{\delta}(1, .gz) = \hat{\delta}(\{1, \texttt{._pdf}, \texttt{._png}, \texttt{._gz}\}, gz) = \hat{\delta}(\{1, \texttt{.g_z}\}, z) = \{1, \texttt{.gz}\}$
- A_{sfx} zamítá řetězec $.c$, protože $\hat{\delta}(1, .c) = \{1\}$ a $\{1\} \cap F = \emptyset$

Příklad: NFA s množinou počátečních stavů

Intuice: nedeterministické automaty jsou v literatuře často definovány s množinou počátečních stavů, tj. mějme NFA $A = (Q, \Sigma, \delta, I, F)$, kde $I \subseteq Q$.

Příklad

NFA $A_1 = (Q_{mod} \cup Q_{\#}, \{a, b\}, \delta_{mod} \cup \delta_{\#}, \{m_0, \#_0\}, \{m_0, \#_2\})$ s množinou počátečních stavů, který rozpoznává jazyk $L_{mod} \cup L_{\#}$.

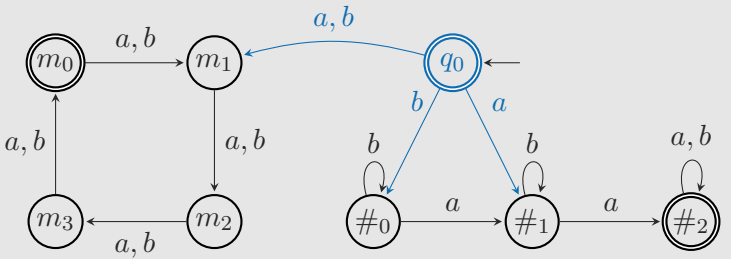


- stavy původních automatů si nejdříve přejmenujeme tak, aby nedocházelo ke kolizím

Příklad: NFA po redukci počátečních stavů

Více počátečních stavů NFA A lze lehce zredukovat na jeden

- přidáme nový počáteční stav q_0 (je zároveň i koncový pokud $\varepsilon \in L(A)$)
- pro každý přechod $\langle i, a, q \rangle \in \delta$, kde $i \in I$ je jeden z původních počátečních stavů, přidáme do δ nový přechod $\langle q_0, a, q \rangle$

Příklad

NFA $A_1' = (Q_{mod} \cup Q_{\#} \cup \{q_0\}, \{a, b\}, \delta', q_0, \{q_0, m_0, \#_2\})$ s jedním počátečním stavem, který rozpoznává jazyk $L_{mod} \cup L_{\#}$: