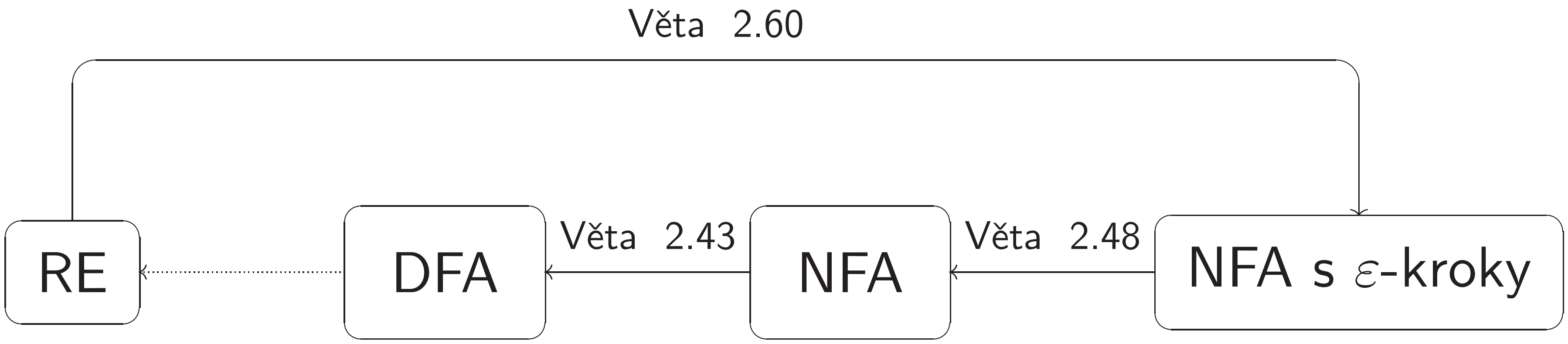


Kleeneho věta 2.63. Libovolný jazyk je popsatelný regulárním výrazem právě když je rozpoznatelný konečným automatem.

Regulární přechodový graf

Definice 2.64.
Regulární přechodový graf $\mathcal{M}$ je pětice $(Q, \Sigma, \delta, I, F)$, kde

- Q je neprázdná konečná množina stavů.
- Σ je vstupní abeceda.
- $\delta : Q \times Q \rightarrow RE(\Sigma)$ je parciální přechodová funkce.
- $I \subseteq Q$ je množina počátečních stavů.
- $F \subseteq Q$ je množina koncových stavů.

Příklad

Slovo $w \in \Sigma^*$ je grafem $\mathcal{M}$ **akceptováno**, právě když

- existuje posloupnost stavů $q_0, \ldots, q_n$, kde $n \geq 1$, $q_0 \in I$, $q_n \in F$
- a $\delta(q_{i-1}, q_i)$ je definováno pro každé $0 < i \leq n$

taková, že

- w lze rozdělit na n částí $w = v_1 \ldots v_n$ tak, že
- $v_i \in L(\delta(q_{i-1}, q_i))$ pro každé $0 < i \leq n$.

Slovo ε je akceptováno také tehdy, je-li $I \cap F \neq \emptyset$.

Převod regulárního přechodového grafu na NFA

Motivace

Věta 2.65. Pro libovolný regulární přechodový graf $\mathcal{M} = (Q, \Sigma, \delta, I, F)$ existuje ekvivalentní NFA $\mathcal{M}'$ s ε-kroky.

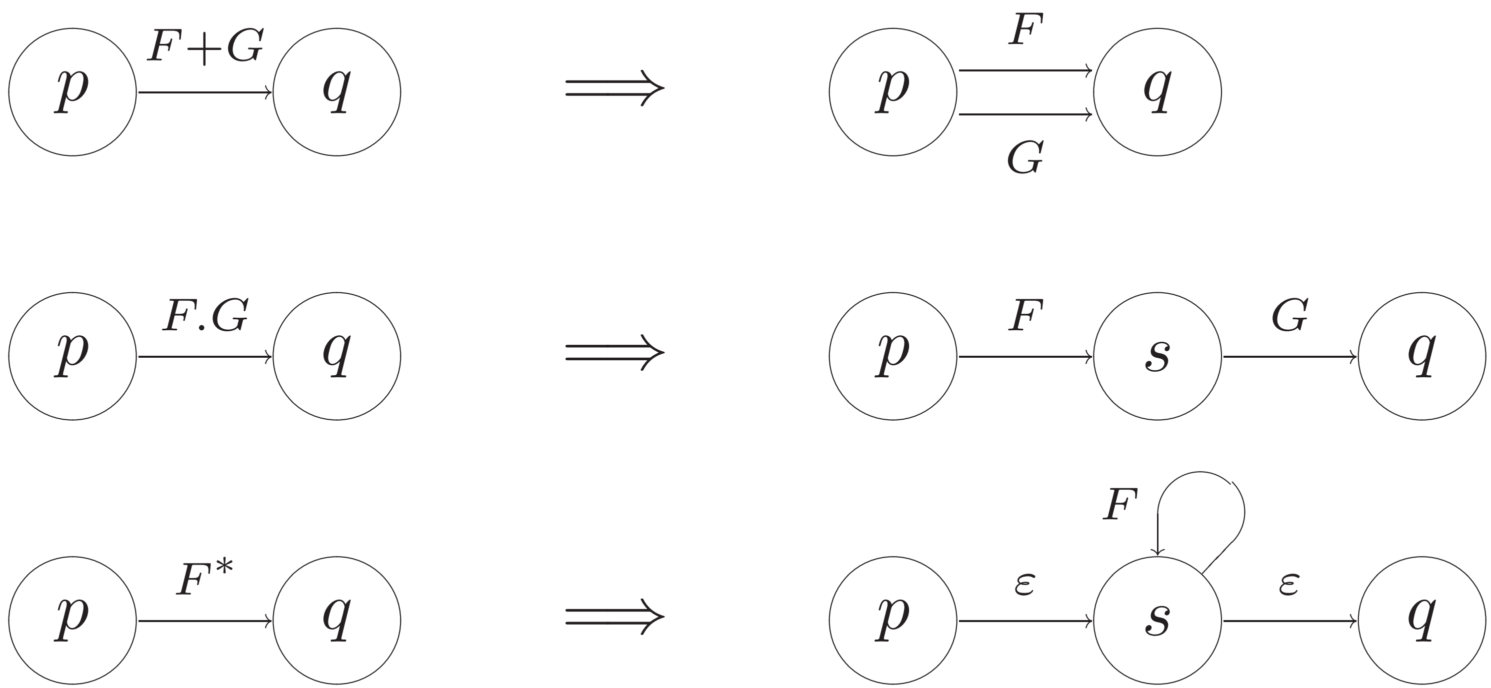
Důkaz. Algoritmus konstrukce NFA $\mathcal{M}'$ s ε-kroky.

Krok 1 Ke grafu $\mathcal{M}$ přidáme nový stav q_0 a hranu $q_0 \xrightarrow{\varepsilon} q$ pro každé $q \in I$. Stav q_0 bude (jediným) počátečním stavem automatu $\mathcal{M}'$, prvky F jeho koncovými stavy.

Krok 2 Opakovaně realizuj kroky **(a)** a **(b)** dokud přechodový graf obsahuje alespoň jednu hranu ohodnocenou symbolem, který nepatří do $\Sigma \cup \{\varepsilon\}$, tedy je tvaru $F + G$, $F.G$, F^* nebo $\emptyset$.

(a) Odstraň všechny hrany, které jsou ohodnoceny symbolem $\emptyset$.

(b) Vyber libovolnou hranu $p \xrightarrow{E} q$, kde $E \notin \Sigma \cup \{\varepsilon\}$, odstraň ji a proveď následující:



Konečnost algoritmu $\mathcal{M}$ obsahuje pouze konečně mnoho hran a každý regulární výraz vznikne aplikací pravidel 1–2 z definice regulárních výrazů (2.58) v konečně mnoha krocích.

Korektnost algoritmu

- Výsledný graf je přechodovým grafem nedeterministického konečného automatu $\mathcal{M}'$ s ε-kroky.
- Přímo z definice regulárního přechodového grafu se snadno ověří, že kroky 1 a 2 popsaného transformačního algoritmu zachovávají ekvivalenci automatů, proto $L(\mathcal{M}) = L(\mathcal{M}')$.

$\square$

Převod DFA na regulární přechodový graf

Motivace

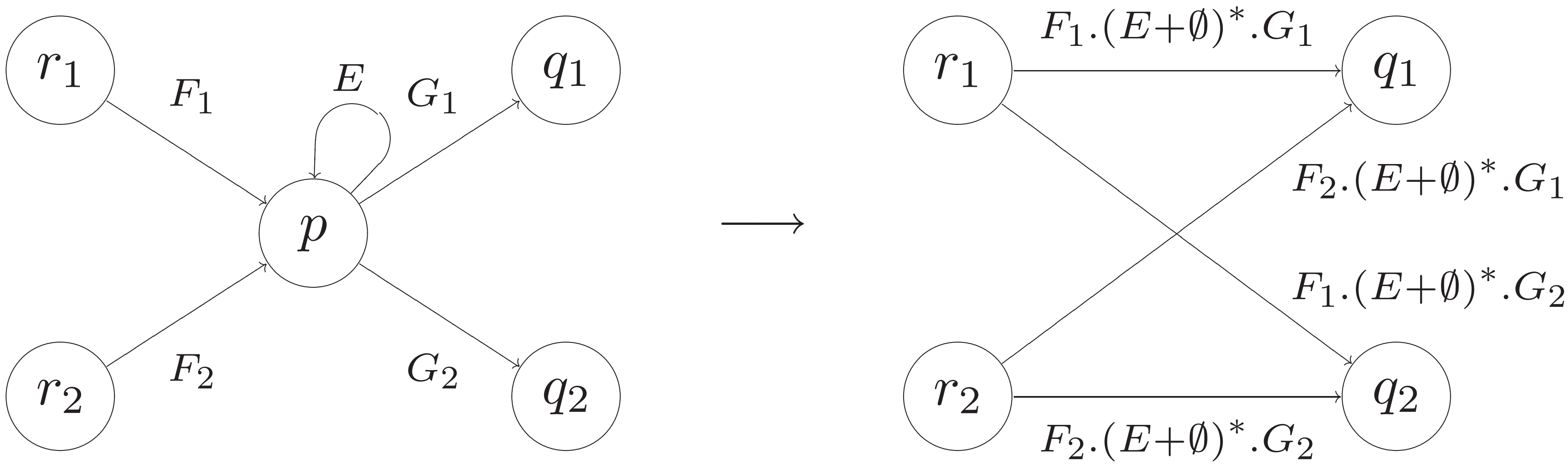
Věta 2.66. Pro každý regulární přechodový graf $\mathcal{M} = (Q, \Sigma, \delta, I, F)$ existuje ekvivalentní regulární přechodový graf $\mathcal{M}' = (\{x, y\}, \Sigma, \delta', \{x\}, \{y\})$, kde δ' může být definováno pouze pro dvojici (x, y).

Důkaz. Algoritmus transformace

Krok 1 Ke grafu $\mathcal{M}$ přidáme nový počáteční stav x a nový koncový stav y. Přidáme také hrany $x \stackrel{\varepsilon}{\to} q$ pro každé $q \in I$ a $r \stackrel{\varepsilon}{\to} y$ pro každé $r \in F$.

Krok 2 Každý stav p různý od x, y nyní odstraníme spolu s hranami, které do p vcházejí nebo z p vycházejí. Pokud do p nevede hrana z jiného uzlu, je nedosažitelný z počátečního stavu. Pokud z p nevede hrana do jiného uzlu, nelze z p dosáhnout koncový stav. V obou případech p odstraníme bez náhrady.

Pokud do p vchází hrana z jiného uzlu a také z p vychází hrana do jiného uzlu, postupujeme následovně:



Po odstranění všech stavů různých od x a y zůstanou tyto dva stavy spolu s (žádnou nebo jednou) hranou z x do y.

Konečnost algoritmu Každým krokem 2 snížíme počet stavů.

Korektnost algoritmu Z definice regulárního přechodového grafu přímo ověříme, že kroky 1 i 2 zachovávají ekvivalenci. $\square$

Ekvivalence konečných automatů a regulárních gramatik

Pojem regulárního jazyka byl definován dvakrát – nejprve pomocí regulární gramatiky a pak ještě jednou pomocí konečného automatu.

Převod regulární gramatiky na konečný automat

Lemma 2.69. Ke každé regulární gramatice $\mathcal{G} = (N, \Sigma, P, S)$ existuje nedeterministický konečný automat $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$ takový, že $L(\mathcal{G}) = L(\mathcal{M})$.

Důkaz.

Konstrukce konečného automatu

- $Q = \{\overline{A} \mid A \in N\} \cup \{q_f\}$, kde $q_f \notin N$.

- $q_0 = \overline{S}$.

- δ je nejmenší funkce $Q \times \Sigma \to 2^Q$ splňující:
  - Pokud $A \to aB$ je pravidlo v P, pak $\overline{B} \in \delta(\overline{A}, a)$.
  - Pokud $A \to a$ je pravidlo v P, kde $a \neq \varepsilon$, pak $q_f \in \delta(\overline{A}, a)$

- $F = \begin{cases} \{\overline{S}, q_f\} & \text{pokud } S \to \varepsilon \text{ je pravidlo v } P, \\ \{q_f\} & \text{jinak.} \end{cases}$

Korektnost Nejprve indukcí vzhledem ke k dokážeme, že pro každé $a_1, \ldots, a_k \in \Sigma$ a $B \in N$ platí

$$S \Rightarrow^* a_1 \ldots a_k B \iff \overline{B} \in \hat{\delta}(\overline{S}, a_1 \ldots a_k).$$

- **Základní krok $k = 0$:** Z definice $\hat{\delta}$ plyne $\hat{\delta}(\overline{S}, \varepsilon) = \{\overline{S}\}$. Proto

$$S \Rightarrow^* B \iff B = S \iff \overline{B} = \overline{S} \iff \overline{B} \in \hat{\delta}(\overline{S}, \varepsilon)$$

- **Indukční krok:**

$S \Rightarrow^* a_1 \ldots a_{k+1} B$

$\iff \exists C \in N$ takové, že $S \Rightarrow^* a_1 \ldots a_k C \Rightarrow a_1 \ldots a_{k+1} B$

$\iff \exists C \in N$ takové, že $\overline{C} \in \hat{\delta}(\overline{S}, a_1 \ldots a_k) \ \wedge \ C \to a_{k+1} B$

$\iff \exists C \in N$ takové, že $\overline{C} \in \hat{\delta}(\overline{S}, a_1 \ldots a_k) \ \wedge \ \overline{B} \in \delta(\overline{C}, a_{k+1})$

$\iff \overline{B} \in \hat{\delta}(\overline{S}, a_1 \ldots a_{k+1})$.

Dokázali jsme: $S \Rightarrow^* a_1 \ldots a_k B \iff \overline{B} \in \hat{\delta}(\overline{S}, a_1 \ldots a_k)$

Ukážeme, že $w \in L(\mathcal{G}) \iff w \in L(\mathcal{M})$:

- $w = \varepsilon$:

$$\varepsilon \in L(\mathcal{G}) \iff S \to \varepsilon \in P \iff \overline{S} \in F \iff \varepsilon \in L(\mathcal{M})$$

- $w = va$, kde $v \in \Sigma^*$, $a \in \Sigma$:

$$\begin{aligned} va \in L(\mathcal{G}) &\iff S \Rightarrow^* vB \Rightarrow va \\ &\iff S \Rightarrow^* vB \ \wedge\ B \to a \in P \\ &\iff \overline{B} \in \hat{\delta}(\overline{S}, v) \ \wedge\ q_f \in \delta(\overline{B}, a) \\ &\iff q_f \in \hat{\delta}(\overline{S}, va) \iff va \in L(\mathcal{M}) \end{aligned}$$

$\square$

Převod konečného automatu na regulární gramatiku

Lemma 2.71 Pro každý konečný automat $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$ existuje regulární gramatika $\mathcal{G} = (N, \Sigma, P, S)$ taková, že $L(\mathcal{M}) = L(\mathcal{G})$.

Důkaz.

Bez újmy na obecnosti předpokládejme, že $\mathcal{M}$ je nedeterministický.

- $N = \{\overline{q} \mid q \in Q\} \cup \{S\}$, kde $S \notin Q$.
- P je nejmenší množina pravidel splňující:
  - Pokud $p \in \delta(q, a)$, je $\overline{q} \to a\overline{p}$ pravidlo v P.
  - Pokud $p \in \delta(q, a)$ a $p \in F$, je $\overline{q} \to a$ pravidlo v P.
  - Pokud $p \in \delta(q_0, a)$, je $S \to a\overline{p}$ pravidlo v P.
  - Pokud $p \in \delta(q_0, a)$ a $p \in F$, je $S \to a$ pravidlo v P.
  - Pokud $q_0 \in F$, je $S \to \varepsilon$ pravidlo v P.

Gramatika $\mathcal{G} = (N, \Sigma, P, S)$ je zřejmě regulární.

Platí: $\hat{\delta}(q_0, a_1 \ldots a_k) \cap F \neq \emptyset$, kde $k \geq 0, a_1, \ldots, a_k \in \Sigma$

$$\iff S \Rightarrow^* a_1 \ldots a_k$$

$\square$

Rozhodnutelné problémy pro třídu reg. jazyků

Regulární jazyk – popsaný některým z uvažovaných formalismů.

Otázky: Máme-li dány konečné automaty $\mathcal{M}$ a $\mathcal{M}'$ nad Σ

ekvivalence: jsou $\mathcal{M}$ a $\mathcal{M}'$ ekvivalentní? (platí $L(\mathcal{M})=L(\mathcal{M}')$?)

inkluze (jazyků): platí $L(\mathcal{M}) \subseteq L(\mathcal{M}')$?

příslušnost (slova k jazyku): je-li dáno $w \in \Sigma^*$, platí $w \in L(\mathcal{M})$?

prázdnost (jazyka): je $L(\mathcal{M}) = \emptyset$?

univerzalita (jazyka): je $L(\mathcal{M}) = \Sigma^*$?

konečnost (jazyka): je $L(\mathcal{G})$ konečný jazyk?

Věta 2.74 Problém **prázdnosti** ($L(\mathcal{M}) \stackrel{?}{=} \emptyset$) a problém **univerzality** ($L(\mathcal{M}) \stackrel{?}{=} \Sigma^*$) jsou rozhodnutelné pro regulární jazyky.

Důkaz. $L(\mathcal{M})$ je prázdný, právě když mezi dosažitelnými stavy automatu $\mathcal{M}$ není žádný koncový stav.
Univerzalita: $L(\mathcal{M}) = \Sigma^* \iff \text{co}\text{–}L(\mathcal{M}) = \emptyset$. $\square$

Věta 2.77 Problém **ekvivalence** je rozhodnutelný pro regulární jazyky.

Důkaz. Pro libovolné L_1, L_2 platí:

$$(L_1 = L_2) \iff (L_1 \cap \text{co–}L_2) \cup (\text{co–}L_1 \cap L_2) = \emptyset.$$

Pro L_1, L_2 zadané automaty lze uvedené operace algoritmicky realizovat.
Alternativně: minimalizace a kanonizace. $\square$

Věta 2.76 Problém, zda $L(\mathcal{M})$ je **konečný**, resp. **nekonečný**, je rozhodnutelný.

Důkaz. L je nekonečný, právě když DFA $\mathcal{M}$ akceptuje alespoň jedno slovo $w \in \Sigma^*$ s vlastností $n \leq |w| < 2n$, kde $n = card(Q)$.

($\Longrightarrow$) L nekonečný, pak existuje $u \in L$ takové, že $|u| \geq n$.
Je-li $|u| < 2n$, jsme hotovi.
Jinak z lemma o vkládání plyne, že $u = xyz$, kde $1 \leq |y| \leq n$ a $xz \in L$.
Platí $|xz| \geq n$. Pokud $|xz| \geq 2n$, celý postup opakujeme.

($\Longleftarrow$) $|w| \geq n$, pak $\mathcal{M}$ na w musí projít dvakrát stejným stavem.
Proto $w = xyz$ tak, že $|y| \geq 1$ a platí $xy^iz \in L$ pro každé $i \in \mathbb{N}_0$ (viz důkaz lemmatu o vkládání), tedy L je nekonečný.

Existenci $w \in L$ takového, že $n \leq |w| < 2n$, lze algoritmicky ověřit (slov je konečně mnoho, "vyzkoušíme" každé z nich). $\square$

Aplikace reg. jazyků a konečných automatů

vyhledávání vzorů (pattern matching) v textu (editory, textové systémy), DNA sekvencích, . . .

Například v Unixu:

`grep` - vyhledávání podle zadaného regulárního výrazu
`egrep` - vyhledávání podle zadaného rozšířeného regulárního výrazu
`fgrep` - vyhledávání podle zadaného řetězce

Zpracování lexikálních jednotek například při automatizované konstrukci překladačů (`lex`, `flex`)

Zpracování obrazů (image processing)

Konečné automaty nad nekonečnými slovy

Specifikace a verifikace konečně stavových systémů

Konečné automaty s výstupem