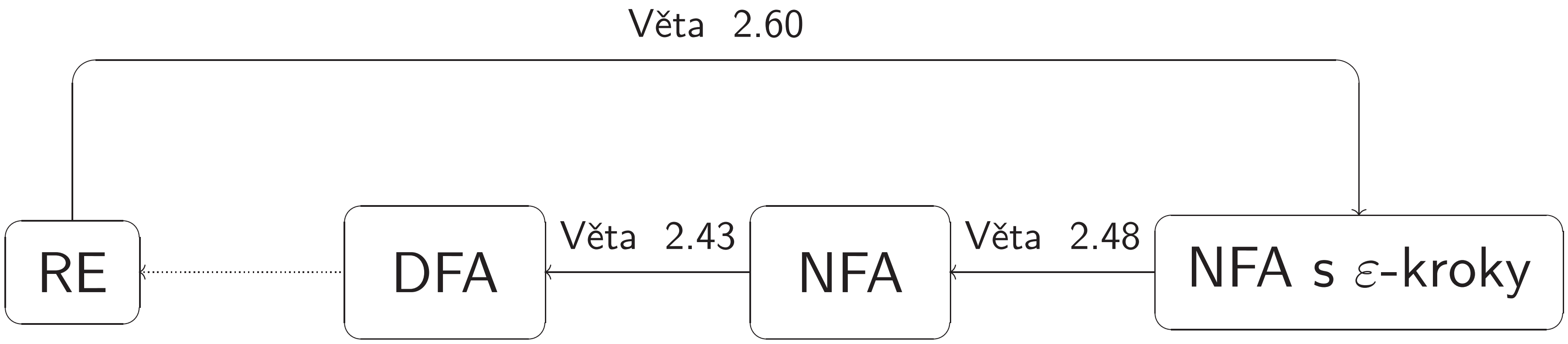


Kleeneho věta 2.63. Libovolný jazyk je popsatelný regulárním výrazem právě když je rozpoznatelný konečným automatem.

Převod DFA na regulární přechodový graf

Motivace

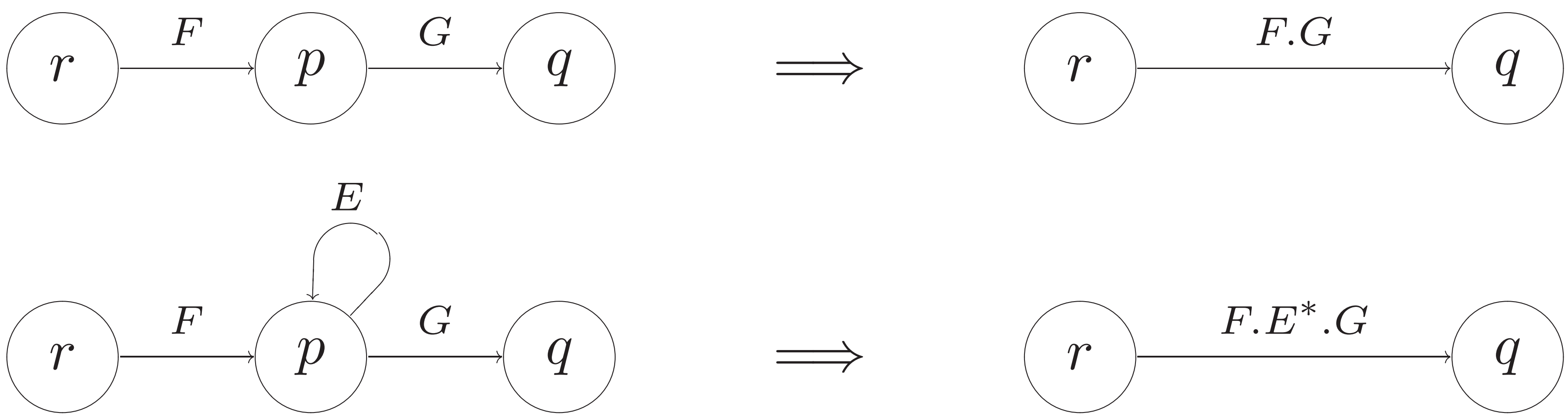
Věta 2.66. Pro každý regulární přechodový graf $\mathcal{M} = (Q, \Sigma, \delta, I, F)$ existuje ekvivalentní regulární přechodový graf $\mathcal{M}' = (\{x, y\}, \Sigma, \delta', \{x\}, \{y\})$, kde δ' může být definováno pouze pro dvojici (x, y).

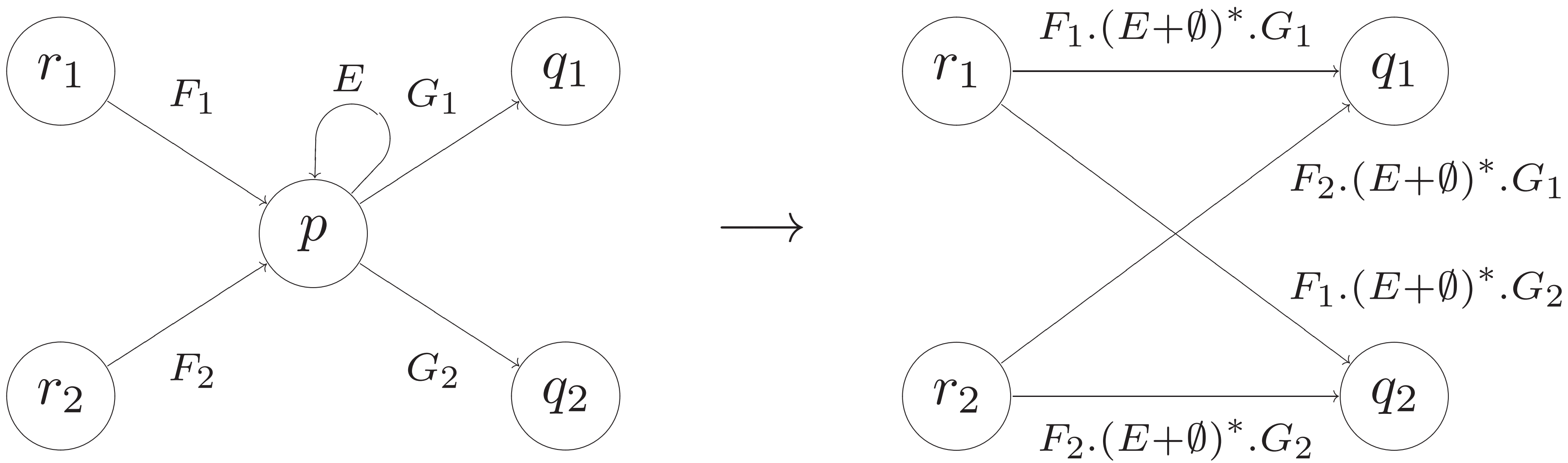
Důkaz. Algoritmus transformace

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.

Pro každou dvojici vstupní hrany vedoucí do p z jiného uzlu a výstupní hrany vedoucí z p do jiného uzlu přidáme přímý přechod. Pak p odstraníme.





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

$$\begin{aligned}
&\iff \exists C \in N \text{ takové, že } S \Rightarrow^* a_1 \ldots a_k C \Rightarrow a_1 \ldots a_{k+1} B \\
&\iff \exists C \in N \text{ takové, že } \overline{C} \in \hat{\delta}(\overline{S}, a_1 \ldots a_k) \ \wedge\ C \to a_{k+1} B \\
&\iff \exists C \in N \text{ takové, že } \overline{C} \in \hat{\delta}(\overline{S}, a_1 \ldots a_k) \ \wedge\ \overline{B} \in \delta(\overline{C}, a_{k+1}) \\
&\iff \overline{B} \in \hat{\delta}(\overline{S}, a_1 \ldots a_{k+1}).
\end{aligned}$$

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

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 $k \geq 0, a_1, \ldots, a_k \in \Sigma$

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

□

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: $L(\mathcal{M}) = \Sigma^* \iff$ co–$L(\mathcal{M}) = \emptyset$. $\square$

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 jazyk L zadaný automatem $\mathcal{M}$ je **konečný**, resp. **nekonečný**, je rozhodnutelný.

Důkaz. Nechť $\mathcal{M}$ je DFA. L je nekonečný právě když $\mathcal{M}$ 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.
Nechť $|u| \geq 2n$. 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

Bezkontextové jazyky

Bezkontextová gramatika (context-free grammar, CFG) $\mathcal{G}$ je čtveřice (N, Σ, P, S), kde

- N je neprázdná konečná množina **neterminálních symbolů**,
- Σ je konečná množina **terminálních symbolů** taková, že $N \cap \Sigma = \emptyset$ (značení: $V = N \cup \Sigma$),
- $S \in N$ je **počáteční neterminál**,
- $P \subseteq N \times V^*$ je konečná množina **pravidel**.

Jazyk je **bezkontextový**, pokud je generovaný nějakou bezkontextovou gramatikou.

Příklad

$\mathcal{G} = (\{E, T, F\}, \{+, *, (,), i\}, P, E)$, kde P obsahuje pravidla

$$\begin{array}{lcl} E & \to & E + T \mid T \\ T & \to & T * F \mid F \\ F & \to & (E) \mid i \end{array}$$

Derivační stromy pro bezkontextové gramatiky

Definice 3.1. Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je CFG.
Strom T nazveme **derivačním stromem** v $\mathcal{G}$ právě když

1. kořen má návěští S, vnitřní uzly mají návěští z N, listy mají navěští z $N \cup \Sigma \cup \{\varepsilon\}$,
2. má-li vnitřní uzel návěští A a jeho všichni synové $n_1, \ldots, n_k$ mají v uspořádání zleva doprava návěští $X_1, \ldots, X_k \in V$, pak $A \to X_1 \ldots X_k \in P$,
3. každý list s návěštím ε je jediným synem svého otce.

Výsledkem derivačního stromu T nazveme slovo vzniklé zřetězením návěští listů v uspořádání zleva doprava.

Vztah mezi derivačními stromy a relací $\Rightarrow^*$

Věta 3.3. Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je CFG. Pak pro libovolné $\alpha \in (N \cup \Sigma)^*$ platí $S \Rightarrow^* \alpha$ právě když v $\mathcal{G}$ existuje derivační strom s výsledkem α.

Důkaz. Označme $\mathcal{G}_A \stackrel{def}{=} (N, \Sigma, P, A)$, kde $A \in N$. Dokážeme, že pro každé $A \in N$ platí

$$A \Rightarrow^* \alpha \iff \text{ v } \mathcal{G}_A \text{ existuje derivační strom s výsledkem } \alpha$$

($\Longleftarrow$) Nechť α je výsledkem derivačního stromu, který má k vnitřích uzlů. Indukcí vzhledem ke k ukážeme, že pak $A \Rightarrow^* \alpha$.

Základní krok $k = 1$:

Indukční krok $k > 1$:

(IP) Tvrzení platí pro stromy s nejvýše $k-1$ vnitřními uzly.

Strom T s k uzly:

- je-li X_i list, označme $\alpha_i = X_i$
- není-li X_i list, pak α_i je výsledkem podstromu T_i s kořenem X_i
- Výsledek T je $\alpha_1 \ldots \alpha_n$.

Platí: $X_i \Rightarrow^* \alpha_i$ (pro X_i, které není listem, podle (IP))

$A \to X_1 \ldots X_n \in P$ (z definice deriv. stromu)

Dostáváme $A \Rightarrow X_1 \ldots X_n \Rightarrow^* \alpha_1 \ldots \alpha_n$.

($\Longrightarrow$) Nechť $A \Rightarrow^* \alpha$. Ukážeme, že v $\mathcal{G}_A$ existuje derivační strom s výsledkem α. Použijeme indukci k délce odvození $A \Rightarrow^* \alpha$.

Základní krok $A \overset{0}{\Rightarrow} \alpha$**:** Pak $\alpha = A$ a odpovídající derivační strom má jen jeden uzel (kořen je list) s označením A.

Indukční krok $A \overset{k+1}{\Rightarrow} \alpha$**,** $k \geq 0$**:**
(IP) Pro každé $B \in N$ platí: pokud $B \Rightarrow^* \beta$ v nejvýše k krocích, pak v $\mathcal{G}_B$ existuje derivační strom s výsledkem β.

$A \overset{k+1}{\Rightarrow} \alpha \quad\Longrightarrow\quad A \Rightarrow X_1 \ldots X_n \overset{k}{\Rightarrow} \alpha_1 \ldots \alpha_n$, kde $X_i \overset{\leq k}{\Rightarrow} \alpha_i$

Konstrukce stromu s výsledkem α:

$\square$

Jednoznačnost derivačních stromů

Derivace je sekvence $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_n$.

Levá (resp. **pravá**) **derivace** je taková derivace, kde každé α_{i+1} vznikne z α_i přepsáním nejlevějšího (resp. nejpravějšího) neterminálu.

Každému derivačnímu stromu odpovídá jediná levá derivace.
Každé levé derivaci odpovídá jediný derivační strom.

Analogicky pro pravou derivaci.

Existuje pro každé $w \in L(\mathcal{G})$ právě jeden derivační strom?

Definice 3.7. CFG $\mathcal{G}$ se nazývá **víceznačná (nejednoznačná)** právě když existuje $w \in L(\mathcal{G})$ mající alespoň dva různé derivační stromy.

V opačném případě říkáme, že $\mathcal{G}$ je **jednoznačná**.

Bezkontextový jazyk L se nazývá **vnitřně (inherentně) víceznačný**, právě když každá bezkontextová gramatika, která jej generuje, je víceznačná.