

# Kanonické tvary bezkontextových gramatik

- redukované bezkontextové gramatiky
- gramatiky bez  $\varepsilon$ -pravidel
- gramatiky bez jednoduchých pravidel
- Chomského normální forma
- gramatiky bez levé rekurze
- Greibachové normální forma

# Chomského normální forma

**Definice 3.19.** Bezkontextová gramatika  $\mathcal{G} = (N, \Sigma, P, S)$  je v **Chomského normální formě** (CNF), právě když  $\mathcal{G}$  je bez  $\varepsilon$ -pravidel a každé pravidlo z  $P$  má jeden z těchto tvarů:

1  $A \rightarrow BC$ , kde  $B, C \in N$

2  $A \rightarrow a$ , kde  $a \in \Sigma$

3  $S \rightarrow \varepsilon$

**Věta 3.21.** Každý bezkontextový jazyk lze generovat bezkontextovou gramatikou v Chomského normální formě.

# Příklad

$\mathcal{G} = (\{S, A, B\}, \{a, b\}, P, S)$ , kde  $P$  obsahuje pravidla

$S \rightarrow AS \mid a$   
 $A \rightarrow AB \mid AA \mid a$   
 $B \rightarrow b$

# Algoritmus transformace do CNF

1  $L = \emptyset$

2  $L \neq \emptyset$

Gramatiku pro  $L$  převedeme na vlastní a bez jednoduchých pravidel.

$$X \rightarrow \varepsilon$$

$$X \rightarrow a$$

$$X \rightarrow A$$

$$X \rightarrow ab$$

$$X \rightarrow aB$$

$$X \rightarrow Ab$$

$$X \rightarrow AB$$

$\vdots$

$$X \rightarrow aBcD$$

# Lemma o vkládání pro bezkontextové jazyky

**Věta 3.24.** Necht'  $L$  je CFL. Pak existují  $p, q \in \mathbb{N}$  (závisející na  $L$ ) taková, že každé slovo  $z \in L$  delší než  $p$  lze psát ve tvaru  $z = uvwxy$ , kde

- alespoň jedno ze slov  $v, x$  je neprázdné (tj.  $vx \neq \varepsilon$ ),
- $|vwx| \leq q$  a
- $uv^iwx^iy \in L$  pro každé  $i \in \mathbb{N}_0$ .

**Poznámka 3.25.** Tvrzení zůstává v platnosti i když namísto konstant  $p, q$  budeme všude psát jen (jedinou) konstantu  $n$ .

# Důkaz Lemmatu o vkládání pro bezkontextové jazyky

Necht'  $L$  je generován gramatikou v CNF.

**délka cesty** z kořene do listu

= počet modrých hran

= počet neterminálů na cestě - 1

**hloubka stromu**

= maximální délka cesty

Derivační strom hloubky  $k$  má max.  $2^k$  listů  $\implies$  slovo délky nejvýše  $2^k$ .

Derivační strom pro slovo delší než  $2^{k-1}$  má cestu délky alespoň  $k$ .

Tato cesta obsahuje alespoň  $k + 1$  neterminálů.

# Důkaz Lemmatu o vkládání pro bezkontextové jazyky

Nechť  $L$  je generován gramatikou  $\mathcal{G} = (N, \Sigma, P, S)$ , která je v CNF.  
Označme  $k = \text{card}(N)$  a položme  $p = 2^{k-1}$ ,  $q = 2^k$ .

Nechť  $z \in L$  je slovo delší než  $p$ . Pak v libovolném derivačním stromu slova  $z$  existuje cesta délky alespoň  $k$ . Zvolme pevně jeden takový strom  $T$  a v něm (libovolnou) nejdelší cestu  $C$ .

Na cestě  $C$  lze zvolit tři uzly  $u_1, u_2, u_3$  s vlastnostmi:

- 1 uzly  $u_1, u_2$  jsou označeny týmž neterminálem, řekněme  $A$
- 2  $u_1$  leží blíže ke kořenu než  $u_2$
- 3  $u_3$  je list
- 4 cesta z  $u_1$  do  $u_3$  má délku nejvýše  $k$





# Použití Lemmatu o vkládání pro bezkontextové jazyky

Lemma o vkládání je implikace  $P \implies Q$ , kde  $P$  je výrok, že  $L$  je CFL a  $Q$  jsou uvedené vlastnosti.

Obměnu Lemmatu o vkládání  $\neg Q \implies \neg P$  lze použít k důkazu, že nějaký jazyk  $L$  **není** CFL — stačí, když ukážeme platnost  $\neg Q$ .

$\neg Q$ :

- 1 Pro libovolnou konstantu  $n \in \mathbb{N}$
- 2 existuje slovo  $z \in L$  delší než  $n$  takové, že
- 3 pro všechny slova  $u, v, w, x, y$  splňující  $z = uvwxy$ ,  $vx \neq \varepsilon$  a  $|vwx| \leq n$
- 4 existuje  $i \in \mathbb{N}_0$  takové, že  $uv^iwx^iy \notin L$ .

# Příklad použití Lemmatu o vkládání

$$L = \{a^i b^i c^i \mid i \geq 1\}$$

- 1 Pro libovolnou konstantu  $n \in \mathbb{N}$
- 2 existuje slovo  $z \in L$  delší než  $n$  takové, že
- 3 pro všechny slova  $u, v, w, x, y$  splňující  $z = uvwxy$ ,  $vx \neq \varepsilon$  a  $|vwx| \leq n$
- 4 existuje  $i \in \mathbb{N}_0$  takové, že  $uv^i wx^i y \notin L$ .

$\implies L$  není CFL.

# Rekursivní neterminály a gramatiky

**Definice 3.28.** Neterminál  $A$  v CFG  $\mathcal{G} = (N, \Sigma, P, S)$  se nazývá **levorekursivní** jestliže v  $\mathcal{G}$  existuje derivace  $A \Rightarrow^+ A\beta$ .

CFG bez levorekursivních neterminálů se nazývá **nelevorekursivní**.

Je-li v CFG pravidlo tvaru  $A \rightarrow A\alpha$ , hovoříme o **přímé levé rekursi** na  $A$ .

**Praktický význam:** některé nástroje pro automatickou tvorbu parserů k zadaným gramatikám vyžadují na vstupu nelevorekursivní gramatiku (např. ANTLR).

# Algoritmus odstranění přímé levé rekurze

Nechť CFG  $\mathcal{G} = (N, \Sigma, P, S)$  je necyklická a bez  $\varepsilon$ -pravidel, v níž všechna  $A$ -pravidla (pravidla mající na levé straně  $A$ ) jsou tvaru

$$A \rightarrow A\alpha_1 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \dots \mid \beta_n,$$

kde každý řetěz  $\beta_i$  začíná symbolem různým od  $A$ .

Nechť  $\mathcal{G}' = (N \cup \{A'\}, \Sigma, P', S)$ , kde  $P'$  obdržíme z  $P$  tak, že všechna výše uvedená pravidla nahradíme pravidly:

$$A \rightarrow \beta_1 \mid \dots \mid \beta_n \mid \beta_1 A' \mid \dots \mid \beta_n A'$$

$$A' \rightarrow \alpha_1 \mid \dots \mid \alpha_m \mid \alpha_1 A' \mid \dots \mid \alpha_m A'$$

Pak  $L(\mathcal{G}) = L(\mathcal{G}')$  a  $\mathcal{G}'$  je necyklická a bez  $\varepsilon$ -pravidel.

# Lemma o substituci

## Lemma 3.20. (o substituci)

Nechť  $\mathcal{G} = (N, \Sigma, P, S)$  je CFG. Nechť  $A \rightarrow \alpha_1 B \alpha_2 \in P$ .

Nechť  $B \rightarrow \beta_1 \mid \dots \mid \beta_r$  jsou všechna pravidla v  $P$  tvaru  $B \rightarrow \alpha$ .

Definujme  $\mathcal{G}' = (N, \Sigma, P', S)$ , kde

$$P' = (P \setminus \{A \rightarrow \alpha_1 B \alpha_2\}) \cup \{A \rightarrow \alpha_1 \beta_1 \alpha_2 \mid \dots \mid \alpha_1 \beta_r \alpha_2\}.$$

Pak  $L(\mathcal{G}) = L(\mathcal{G}')$ .

# Příklad

$A \rightarrow Bd \mid c$

$B \rightarrow Bdd \mid Ccc \mid aAd$

$C \rightarrow Aa$

# Algoritmus odstranění levé rekurze

**Vstup:** Vlastní CFG  $\mathcal{G} = (N, \Sigma, P, S)$

**Výstup:** Ekvivalentní nelevorekursivní gramatika bez  $\varepsilon$ -pravidel

- 1: Uspořádej libovolně  $N$ ,  $N = \{A_1, \dots, A_n\}$
- 2: **for**  $i \leftarrow 1$  **to**  $n$  **do**
- 3:     **for**  $j \leftarrow 1$  **to**  $i - 1$  **do**
- 4:         **for all** pravidlo tvaru  $A_i \rightarrow A_j\alpha$  **do**
- 5:             přidej pravidla  $A_i \rightarrow \beta_1\alpha \mid \dots \mid \beta_k\alpha$
- 6:             (kde  $A_j \rightarrow \beta_1 \mid \dots \mid \beta_k$  jsou všechna pravidla pro  $A_j$ )
- 7:             vypuť pravidlo  $A_i \rightarrow A_j\alpha$
- 8:         **end for**
- 9:     **end for**
- 10:     odstraň případnou přímou levou rekursi na  $A_i$
- 11: **end for**



# Korektnost algoritmu

Konečnost.

**Ekvivalence gramatik:** Všechny úpravy jsou dle Lemmatu o substituci nebo odstraňují přímou levou rekursi.

**Výsledná gramatika je nelevorekursivní:**

- 1 po  $i$ -té iteraci vnějšího cyklu začíná každé  $A_i$ -pravidlo buď terminálem nebo neterminálem  $A_k$ , kde  $k > i$ .
- 2 po  $j$ -té iteraci vnitřního cyklu začíná každé  $A_i$ -pravidlo buď terminálem nebo neterminálem  $A_k$ , kde  $k > j$ .

**Výsledná gramatika je bez  $\varepsilon$ -pravidel.**

# Greibachové normální forma

**Definice 3.33.** Bezkontextová gramatika  $\mathcal{G} = (N, \Sigma, P, S)$  je v **Greibachové normální formě** (GNF), právě když

- $\mathcal{G}$  je bez  $\varepsilon$ -pravidel a
- každé pravidlo z  $P$  je tvaru  $A \rightarrow a\alpha$ , kde  $a \in \Sigma$  a  $\alpha \in N^*$  (s případnou výjimkou pravidla  $S \rightarrow \varepsilon$ ).

**Věta 3.34.** Každý bezkontextový jazyk lze generovat bezkontextovou gramatikou v Greibachové normální formě.

# Zásobníkové automaty

# Definice zásobníkového automatu

**Definice 3.36. Nedeterministický zásobníkový automat** (PushDown Automaton, PDA) je sedmice  $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ , kde

- $Q$  je konečná množina, jejíž prvky nazýváme **stavy**,
- $\Sigma$  je konečná množina, tzv. **vstupní abeceda**,
- $\Gamma$  je konečná množina, tzv. **zásobníková abeceda**,
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \mathcal{P}_{Fin}(Q \times \Gamma^*)$ , tzv. (parciální) **přechodová funkce**<sup>1</sup>,
- $q_0 \in Q$  je **počáteční stav**,
- $Z_0 \in \Gamma$  je **počáteční symbol v zásobníku**,
- $F \subseteq Q$  je množina **koncových stavů**.

---

<sup>1</sup>Zápis  $\mathcal{P}_{Fin}(Q \times \Gamma^*)$  značí množinu všech **konečných** podmnožin množiny  $Q \times \Gamma^*$ .

# Výpočet zásobníkového automatu

**Definice 3.37.** Necht'  $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$  je PDA.

**Konfigurací** nazveme libovolný prvek  $(p, w, \alpha) \in Q \times \Sigma^* \times \Gamma^*$ .

Na množině všech konfigurací automatu  $\mathcal{M}$  definujeme binární relaci

**krok výpočtu**  $\vdash_{\mathcal{M}}$  takto:

$$(p, aw, Z\alpha) \vdash_{\mathcal{M}} (q, w, \gamma\alpha) \stackrel{\text{def}}{\iff} \exists(q, \gamma) \in \delta(p, a, Z) \text{ pro } a \in \Sigma \cup \{\varepsilon\}$$

Reflexivní a tranzitivní uzávěr relace  $\vdash_{\mathcal{M}}$  značíme  $\vdash_{\mathcal{M}}^*$ .

Je-li  $\mathcal{M}$  zřejmý z kontextu, píšeme pouze  $\vdash$  resp.  $\vdash^*$ .

# Akceptující výpočet zásobníkového automatu

## Definice 3.37. (pokračování)

**Jazyk akceptovaný PDA  $\mathcal{M}$  koncovým stavem** definujeme jako

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q_f, \varepsilon, \alpha), \text{ kde } q_f \in F, \alpha \in \Gamma^*\}$$

a jazyk akceptovaný PDA  $\mathcal{M}$  **prázdným zásobníkem** definujeme jako

$$L_e(\mathcal{M}) = \{w \in \Sigma^* \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon), \text{ kde } q \in Q\}.$$