Kanonické tvary bezkontextových gramatik

- redukované bezkontextové gramatiky
- gramatiky bez ε-pravidel
- gramatiky bez jednoduchých pravidel
- vlastní gramatiky
- 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 ε-pravidel a každé pravidlo z P má jeden z těchto tvarů:

1. $A \to BC$, kde $B, C \in N$
2. $A \to a$, kde $a \in \Sigma$
3. $S \to \varepsilon$

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

$$
\begin{aligned}
S &\to AS \mid a \\
A &\to AB \mid AA \mid a \\
B &\to b
\end{aligned}
$$

Algoritmus transformace do CNF

1. $L = \emptyset$
2. $L \neq \emptyset$
   Gramatiku pro L převedeme na vlastní a bez jednoduchých pravidel.
   $$\begin{aligned} X &\to \varepsilon \\ X &\to a \\ X &\to A \\ X &\to ab \\ X &\to aB \\ X &\to Ab \\ X &\to AB \\ &\vdots \\ X &\to aBcD \end{aligned}$$

Lemma o vkládání pro bezkontextové jazyky

Věta 3.24. Nechť 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

Nechť 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 $\Longrightarrow$ 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 $k = card(N)$ 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 $uv^iwx^iy \notin L$.

$\Longrightarrow$ 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 $A \to A\alpha$, 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 ε-pravidel, v níž všechna A-pravidla (pravidla mající na levé straně A) jsou tvaru

$$A \to A\alpha_1 \mid \ldots \mid A\alpha_m \mid \beta_1 \mid \ldots \mid \beta_n,$$

kde každý řetěz β_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:

$$A \to \beta_1 \mid \ldots \mid \beta_n \mid \beta_1 A' \mid \ldots \mid \beta_n A'$$

$$A' \to \alpha_1 \mid \ldots \mid \alpha_m \mid \alpha_1 A' \mid \ldots \mid \alpha_m A'$$

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

Lemma o substituci

Lemma 3.20. (o substituci)
Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je CFG. Nechť $A \to \alpha_1 B \alpha_2 \in P$.
Nechť $B \to \beta_1 \mid \ldots \mid \beta_r$ jsou všechna pravidla v P tvaru $B \to \alpha$.

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

$$P' = (P \smallsetminus \{A \to \alpha_1 B \alpha_2\}) \ \cup \ \{A \to \alpha_1 \beta_1 \alpha_2 \mid \ldots \mid \alpha_1 \beta_r \alpha_2\}.$$

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

Příklad

$$
\begin{aligned}
A &\to Bd \mid c \\
B &\to Bdd \mid Ccc \mid aAd \\
C &\to Aa
\end{aligned}
$$

Algoritmus odstranění levé rekurze

Vstup: Vlastní CFG $\mathcal{G} = (N, \Sigma, P, S)$
Výstup: Ekvivalentní nelevorekursivní gramatika bez ε-pravidel

Uspořádej libovolně N, $N = \{A_1, \dots, A_n\}$
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** $i - 1$ **do**
        for all pravidlo tvaru $A_i \to A_j\alpha$ **do**
            přidej pravidla $A_i \to \beta_1\alpha \mid \dots \mid \beta_k\alpha$
            (kde $A_j \to \beta_1 \mid \dots \mid \beta_k$ jsou všechna pravidla pro A_j)
            vypusť pravidlo $A_i \to A_j\alpha$
        end for
    end for
    odstraň případnou přímou levou rekursi na A_i
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 ε-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 ε-pravidel a
- každé pravidlo z P je tvaru $A \to a\alpha$, kde $a \in \Sigma$ a $\alpha \in N^*$ (s případnou vyjímkou pravidla $S \to \varepsilon$).

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**,
- Σ je konečná množina, tzv. **vstupní abeceda**,
- Γ je konečná množina, tzv. **zásobníková abeceda**,
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}_{Fin}(Q \times \Gamma^*)$, tzv. (parciální) **přechodová funkce**[1],
- $q_0 \in Q$ je **počáteční stav**,
- $Z_0 \in \Gamma$ je **počáteční symbol v zásobníku**,
- $F \subseteq Q$ je množina **koncových stavů**.

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

Výpočet zásobníkového automatu

Definice 3.37. Nechť $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ 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:

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

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

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