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.

Motivační příklad

$$
\begin{aligned}
A &\rightarrow BB \mid aAa \mid b \\
B &\rightarrow bBa \mid bAb
\end{aligned}
$$

Lemma o substituci (pro připomenutí)

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

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**.

Převod bezkontextové gramatiky $\mathcal{G}$ do GNF:

$L(\mathcal{G})$ je prázdný: zřejmé $(S \to aS)$

$L(\mathcal{G})$ je neprázdný:
1. z vlastní gramatiky eliminujeme levou rekurzi
2. pak převedeme do GNF

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.

Příklad

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

Algoritmus odstranění levé rekurze

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

Výstup: Ekvivalentní nelevorekursivní gramatika bez ε-pravidel

Uspořádej libovolně N, $N = \{A_1, \ldots, A_n\}$
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** $i - 1$ **do**
        foreach pravidlo tvaru $A_i \to A_j\alpha$ **do**
            přidej pravidla $A_i \to \beta_1\alpha \mid \ldots \mid \beta_k\alpha$
            (kde $A_j \to \beta_1 \mid \ldots \mid \beta_k$ jsou všechna A_j-pravidla)
            vypusť pravidlo $A_i \to A_j\alpha$
        od
    od
    odstraň případnou přímou levou rekurzi na A_i
od

Korektnost algoritmu

Konečnost.

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

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.

Příklad

$$
\begin{aligned}
A &\rightarrow Ba \mid Db \mid c \\
B &\rightarrow CC \\
C &\rightarrow aE \\
D &\rightarrow CDa \mid Eb \\
E &\rightarrow bb
\end{aligned}
$$

Algoritmus transformace do GNF

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

najdi lineární uspořádání $\prec$ splňující $(A \to B\alpha) \in P \implies A \prec B$
Označme $N = \{A_1, \ldots, A_n \mid A_{i-1} \prec A_i, 1 < i \leq n\}$
for $i \leftarrow n - 1$ **downto** 1 **do**
    foreach pravidlo tvaru $A_i \to A_j\alpha$, kde $j > i$ **do**
        přidej pravidlo $A_i \to \beta_1\alpha \mid \ldots \mid \beta_k\alpha$
        (kde $A_j \to \beta_1 \mid \ldots \mid \beta_k$ jsou všechna A_j-pravidla)
        vypusť pravidlo $A_i \to A_j\alpha$
    od
od
nahraď potřebné terminály novými neterminály
a přidej příslušná pravidla

Korektnost algoritmu

Konečnost.

Ekvivalence gramatik.

Výsledná gramatika je v GNF.

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

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