Kanonické tvary bezkontextových gramatik

- redukované bezkontextové gramatiky
- gramatiky bez ε-pravidel
- gramatiky bez jednoduchých pravidel
- gramatiky bez levé rekurze
- Chomského normální forma
- Greibachové normální forma

Redukované bezkontextové gramatiky

Definice 3.7 Symbol $X \in N \cup \Sigma$ je **nepoužitelný** v CFG $\mathcal{G} = (N, \Sigma, P, S)$ právě když v $\mathcal{G}$ neexistuje derivace tvaru

$$S \Rightarrow^* wXy \Rightarrow^* wxy$$

pro žádné $w, x, y \in \Sigma^*$. Řekneme, že $\mathcal{G}$ je **redukovaná**, jestliže neobsahuje žádné nepoužitelné symboly.

X je **nepoužitelý typu I** **(tj. nenormovaný)** $\iff$ neexistuje $w \in \Sigma^*$ splňující $X \Rightarrow^* w$

X je **nepoužitelý typu II** **(tj. nedosažitelný)** $\iff$ neexistují $\alpha, \beta \in (N \cup \Sigma)^*$ splňující $S \Rightarrow^* \alpha X \beta$

Nalezení nepoužitelných symbolů typu I (neexistuje $w \in \Sigma^*$: $A \Rightarrow^* w$)

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

Výstup: $N_e = \{A \mid \exists w \in \Sigma^*.\ A \Rightarrow^* w\}$ (normované neterminály)

$i := 0;\ N_0 := \emptyset$

repeat $i := i + 1$

$\quad N_i := N_{i-1} \cup \{A \mid A \to \alpha \in P,\ \alpha \in (N_{i-1} \cup \Sigma)^*\}$

until $N_i = N_{i-1}$

$N_e := N_i$

Korektnost algoritmu

Konečnost.

Správnost výsledku: Dokážeme $A \in N_e \iff \exists w \in \Sigma^*.\ A \Rightarrow^* w$.

($\Longrightarrow$) Indukcí k i dokážeme $A \in N_i \implies \exists w \in \Sigma^*.\ A \Rightarrow^* w$.

Základní krok $i = 0$**:** Platí triviálně, protože $N_0 = \emptyset$.

Indukční krok: (IP) Tvrzení platí pro i. Dokážeme pro $i + 1$.

- $A \in N_i$. Tvrzení plyne z (IP).
- $A \in N_{i+1} \smallsetminus N_i$. Pak existuje $A \to X_1 \dots X_k \in P$, kde každé X_j je terminál nebo neterminál patřící do N_i. Podle (IP) existuje w_j tak, že $X_j \Rightarrow^* w_j$. Tedy $A \Rightarrow X_1 \dots X_k \Rightarrow^* w_1 X_2 \dots X_k \Rightarrow^* \dots \Rightarrow^* w_1 \dots w_k$, kde $w_1 \dots w_k \in \Sigma^*$.

($\Longleftarrow$) Indukcí k n dokážeme

$$A \stackrel{n}{\Rightarrow} w, w \in \Sigma^* \implies A \in N_i \text{ pro nějaké } i.$$

Základní krok $n = 1$**:** $A \to w \in P$ okamžitě dává $i = 1$.

Indukční krok: (IP) Předpokládejme, že dokazované tvrzení platí pro všechna $n' \leq n$.

Nechť $A \stackrel{n+1}{\Rightarrow} w$. $A \Rightarrow X_1 \ldots X_k \stackrel{n}{\Rightarrow} w$, kde $X_j \stackrel{n_j}{\Rightarrow} w_j$ a $n_j \leq n$.
Pokud $X_j \in N$, pak podle (IP) $X_j \in N_{i_j}$ pro nějaké i_j.
Pokud $X_j \in \Sigma$, klademe $i_j = 0$.
Položme $i = 1 + \max\{i_1, \ldots, i_k\}$. Pak zřejmě $A \in N_i$.

Důsledek 3.10. Existuje algoritmus, který pro libovolnou danou CFG $\mathcal{G}$ rozhoduje, zda $L(\mathcal{G}) = \emptyset$.

Důkaz. Stačí ověřit, zda $S \notin N_e$. □

Věta. Nechť $\mathcal{G} = (N, \Sigma, P, S)$ je CFG taková, že $L(\mathcal{G}) \neq \emptyset$. Pak existuje ekvivalentní CFG $\mathcal{G}'$ bez nepoužitelných neterminálů typu I.

Důkaz. Stačí spočítat množinu N_e a položit $\mathcal{G}' = (N_e, \Sigma, P', S)$, kde $P' = P \cap N_e \times (N_e \cup \Sigma)^*$. □

Nalezení nepoužitelných symbolů typu II (neexistují $\alpha, \beta \in (N \cup \Sigma)^*$: $S \Rightarrow^* \alpha X \beta$)

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

Výstup: CFG $\mathcal{G}' = (N', \Sigma', P', S)$ bez nedosažitelných symbolů splňující $L(\mathcal{G}) = L(\mathcal{G}')$

$i := 0;\ V_i := \{S\}$
repeat $i := i + 1$
$\quad V_i := V_{i-1} \cup \{X \in N \cup \Sigma \mid \exists A \in V_{i-1} \,.\, A \to \alpha' X \beta' \in P\}$
until $V_i = V_{i-1}$
$N' := N \cap V_i;\ \Sigma' := \Sigma \cap V_i;\ P' := P \cap (V_i \times V_i^*)$

Korektnost: $X \in N' \cup \Sigma' \iff \exists \alpha, \beta \in (N' \cup \Sigma')^* \,.\, S \Rightarrow^* \alpha X \beta$

Příklad

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

$$
\begin{aligned}
S &\rightarrow aSb \mid c \mid aB \\
A &\rightarrow dA \mid d \\
B &\rightarrow eB
\end{aligned}
$$

Eliminace nepoužitelných symbolů

Věta 3.11. Každý neprázdný bezkontextový jazyk L je generován nějakou redukovanou CFG.

Důkaz. Nechť L je generován nějakou CFG $\mathcal{G}$.

Krok 1. Z $\mathcal{G}$ odstraníme symboly typu I *(výsledek označme $\mathcal{G}_1$)*.

Krok 2. Z $\mathcal{G}_1$ odstraníme symboly typu II *(výsledek označme $\mathcal{G}_2$)*.

Korektnost: Sporem dokážeme, že $\mathcal{G}_2$ je redukovaná CFG. Předpokládejme, že $\mathcal{G}_2$ má nepoužitelný symbol X.

- v $\mathcal{G}_2$ existuje derivace $S \Rightarrow^*_{\mathcal{G}_2} \alpha X \beta$
- všechny symboly z $\mathcal{G}_2$ jsou též v $\mathcal{G}_1$
- pro nějaký terminální řetěz w platí $S \Rightarrow^*_{\mathcal{G}_2} \alpha X \beta \Rightarrow^*_{\mathcal{G}_1} w$
- žádný symbol z derivace $\alpha X \beta \Rightarrow^*_{\mathcal{G}_1} w$ není krokem 2 eliminován a proto $\alpha X \beta \Rightarrow^*_{\mathcal{G}_2} w$

Víme tedy, že existuje derivace $S \Rightarrow^*_{\mathcal{G}_2} \alpha X \beta \Rightarrow^*_{\mathcal{G}_2} w$, kde w je terminální řetěz. To je ve sporu s předpokladem, že X je v $\mathcal{G}_2$ nepoužitelný. $\square$

ε-pravidla

Definice 3.13. Řekneme, že CFG $\mathcal{G} = (N, \Sigma, P, S)$ je **bez ε-pravidel** právě když buď

1. P neobsahuje žádné ε-pravidlo (tj. pravidlo tvaru $A \to \varepsilon$) nebo
2. v P existuje právě jedno ε-pravidlo $S \to \varepsilon$ a S se nevyskytuje na pravé straně žádného pravidla z P.

Příklad

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

$$
\begin{aligned}
S &\rightarrow aAbBc \\
A &\rightarrow BB \mid a \mid \varepsilon \\
B &\rightarrow AcA \mid b
\end{aligned}
$$

Algoritmus pro odstranění ε-pravidel

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

Výstup: CFG $\mathcal{G}' = (N', \Sigma, P', S')$ bez ε-pravidel splňující $L(\mathcal{G}){=}L(\mathcal{G}')$

Zkonstruuj $N_\varepsilon = \{A \in N \mid A \Rightarrow^* \varepsilon\}$
Množinu pravidel P' zkonstruuj takto:
foreach $A \to X_1 \dots X_n \in P$ **do**
  přidej do P' všechna pravidla tvaru $A \to \alpha_1 \dots \alpha_n$ splňující
    (a) pokud $X_i \notin N_\varepsilon$ pak $\alpha_i = X_i$
    (b) pokud $X_i \in N_\varepsilon$ pak α_i je buď X_i, nebo ε
    (c) ne všechna α_i jsou ε
od
if $S \in N_\varepsilon$ **then** přidej do P' pravidla $S' \to S \mid \varepsilon$ $(S' \notin N \cup \Sigma)$;
  $N' := N \cup \{S'\}$
  else $N' := N;\ S' := S$ **fi**

Příklad

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

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

Korektnost algoritmu

Konečnost.

Výsledná gramatika je bez ε-pravidel.

Ekvivalence gramatik.

Jednoduchá pravidla

Jednoduchým pravidlem nazývame každé pravidlo tvaru $A \to B$, kde $A, B \in N$.

$$\begin{array}{lcl} S & \to & aAbBc \\ A & \to & aA \mid B \mid a \\ B & \to & bB \mid b \end{array}$$

Algoritmus pro odstranění jednoduchých pravidel

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

Výstup: CFG $\mathcal{G}' = (N, \Sigma, P', S)$ bez jednoduchých a ε-pravidel, kde $L(\mathcal{G}) = L(\mathcal{G}')$

foreach $A \in N$ **do**
  $i := 0; \quad N_i := \{A\}$
  repeat $i := i + 1$
    $N_i := N_{i-1} \cup \{C \mid B \to C \in P, B \in N_{i-1}\}$
   until $N_i = N_{i-1}$
  $N_A := N_i$
od
$P' := \emptyset$
foreach $A \in N$ **do**
  $P' := P' \cup \{A \to \alpha \mid B \in N_A \wedge B \to \alpha \in P$ není jednoduché$\}$
od

Příklad

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

$$
\begin{array}{lcl}
S & \rightarrow & ABC \\
A & \rightarrow & aA \mid B \mid a \\
B & \rightarrow & bB \mid A \\
C & \rightarrow & cC \mid A
\end{array}
$$

Korektnost algoritmu

Konečnost.

Výsledná gramatika neobsahuje jednoduchá pravidla.

Ekvivalence gramatik:

$L(\mathcal{G}') \subseteq L(\mathcal{G})$ Nechť $w \in L(\mathcal{G}')$, pak existuje derivace

$S = \alpha_0 \Rightarrow_{\mathcal{G}'} \alpha_1 \Rightarrow_{\mathcal{G}'} \ldots \Rightarrow_{\mathcal{G}'} \alpha_n = w$.

Pokud bylo při kroku $\alpha_i \Rightarrow_{\mathcal{G}'} \alpha_{i+1}$ použito pravidlo $A \to \beta$, pak existuje nějaké $B \in N_A$ takové, že v $\mathcal{G}$ platí $A \Rightarrow^* B$ a $B \to \beta$. Tedy v $\mathcal{G}$ platí $A \Rightarrow^* \beta$ a $\alpha_i \Rightarrow^* \alpha_{i+1}$.

$L(\mathcal{G}) \subseteq L(\mathcal{G}')$ Nechť $w \in L(\mathcal{G})$, pak existuje levá derivace

$S = \alpha_0 \Rightarrow_{\mathcal{G}} \alpha_1 \Rightarrow_{\mathcal{G}} \ldots \Rightarrow_{\mathcal{G}} \alpha_n = w$.

Tu lze rozdělit na úseky tak, že v celém úseku se použila pouze jednoduchá pravidla anebo žádné jednoduché pravidlo. Úseky s jednoduchými pravidly lze nahradit.

Vlastní bezkontextová gramatika

Definice 3.17. CFG $\mathcal{G} = (N, \Sigma, P, S)$ se nazývá **necyklická**, právě když neexistuje $A \in N$ takový, že $A \Rightarrow^+ A$.

$\mathcal{G}$ se nazývá **vlastní**, právě když je bez nepoužitelných symbolů, bez ε-pravidel a necyklická.

Věta 3.18. Ke každému neprázdnému bezkontextovému jazyku existuje **vlastní** bezkontextová gramatika, která jej generuje.

Důkaz. Z bezkontextové gramatiky pro neprázdný jazyk odstraníme ε-pravidla a jednoduchá pravidla. Odstraněním nepoužitelných symbolů pak získáme vlastní gramatiku. $\square$

Chomského normální forma

Definice 3.19. Bezkontextová gramatika $\mathcal{G} = (N, \Sigma, P, S)$ je v **Chomského normální formě** (CNF) $\overset{def}{\Longleftrightarrow}$ $\mathcal{G}$ 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

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

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

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

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

Algoritmus transformace do CNF

1. $L = \emptyset$
2. $L \neq \emptyset$
   Gramatiku pro L převedeme na vlastní a bez jednoduchých pravidel.

$$
\begin{aligned}
X &\rightarrow \varepsilon \\
X &\rightarrow a \\
X &\rightarrow A \\
X &\rightarrow ab \\
X &\rightarrow aB \\
X &\rightarrow Ab \\
X &\rightarrow AB \\
&\vdots \\
X &\rightarrow aBcD
\end{aligned}
$$

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 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 a
4. cesta z u_1 do u_3 má délku nejvýše k.

Použití Lemmatu o vkládání pro bezkontextové jazyky

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

Obměnu Lemmatu o vkládání $\neg Q \Longrightarrow \neg P$ 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.