

Bezkontextové jazyky

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

- N je neprázdňá konečňá množina **neterminálních symbolů**,
- Σ je konečňá 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 konečňá množina **pravidel**.

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

Příklad

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

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid i$$

Derivační stromy pro bezkontextové gramatiky

Definice 3.1. Necht' $\mathcal{G} = (N, \Sigma, P, S)$ 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 návěští z $N \cup \Sigma \cup \{\varepsilon\}$,
- 2 má-li vnitřní uzel návěští A a jeho všichni synové n_1, \dots, n_k mají v uspořádání zleva doprava návěští $X_1, \dots, X_k \in N \cup \Sigma \cup \{\varepsilon\}$, pak $A \rightarrow X_1 \dots X_k \in P$,
- 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. Necht' $\mathcal{G} = (N, \Sigma, P, S)$ 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

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

(\Leftarrow) Necht' 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 $\alpha_1 \dots \alpha_n$

Platí: $X_i \Rightarrow^* \alpha_i$ (pro X_i , které není listem, podle (IP))
 $A \rightarrow X_1 \dots X_n \in P$ (z definice derivačního stromu)

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

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

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

Indukční krok $A \xRightarrow{k+1} \alpha$, $k \geq 0$:

(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 .

$$A \xRightarrow{k+1} \alpha \implies A \Rightarrow X_1 \dots X_n \xRightarrow{k} \alpha_1 \dots \alpha_n, \text{ kde } X_i \xRightarrow{\leq k} \alpha_i$$

Konstrukce stromu s výsledkem α :



Jednoznačnost derivačních stromů

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

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 3.5. 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.

Kanonické tvary bezkontextových gramatik

- redukované bezkontextové gramatiky
- gramatiky bez ε -pravidel
- gramatiky bez jednoduchých pravidel
- Chomského normální forma
- gramatiky bez levé rekurze
- 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 **nepoužitelný typu I** \iff neexistuje $w \in \Sigma^*$
(tj. **nenormovaný**) splňující $X \Rightarrow^* w$

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

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

2: **repeat**

3: $i := i + 1$

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

5: **until** $N_i = N_{i-1}$

6: $N_e := N_i$

Korektnost algoritmu

Konečnost.

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

(\implies) 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).
- $A \in N_{i+1} \setminus N_i$. Pak existuje $A \rightarrow X_1 \dots X_k \in P$, 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 $w_1 \dots w_k \in \Sigma^*$.

(\Leftarrow) Indukcí k n dokážeme

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

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

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

Nechť $A \xRightarrow{n+1} w$. Pak $A \Rightarrow X_1 \dots X_k \xRightarrow{n} w$, kde $X_j \xRightarrow{n_j} w_j$ 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 $i = 1 + \max\{i_1, \dots, i_k\}$. 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. Necht' $\mathcal{G} = (N, \Sigma, P, S)$ 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 $L(\mathcal{G}) = L(\mathcal{G}')$

1: $i := 0; V_0 := \{S\}$

2: **repeat**

3: $i := i + 1$

4: $V_i := V_{i-1} \cup \{X \in N \cup \Sigma \mid \exists A \in V_{i-1}. A \rightarrow \alpha' X \beta' \in P\}$

5: **until** $V_i = V_{i-1}$

6: $N' := N \cap V_i; \Sigma' := \Sigma \cap V_i; P' := P \cap (V_i \times V_i^*)$

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

$S \rightarrow aSb \mid c \mid aB$

$A \rightarrow dA \mid d$

$B \rightarrow eB$

Eliminace nepoužitelných symbolů

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

Důkaz. Necht' 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).

Platí $L(\mathcal{G}) = L(\mathcal{G}_1) = L(\mathcal{G}_2)$.

Korektnost: Dokážeme, že \mathcal{G}_2 je redukovaná CFG.

Nechť X je libovolný symbol z \mathcal{G}_2 .

- 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. Tudíž X není nepoužitelný v \mathcal{G}_2 . □

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 $A \rightarrow \varepsilon$) nebo
- 2 v P existuje právě jedno ε -pravidlo $S \rightarrow \varepsilon$ 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

$S \rightarrow aAbBc$

$A \rightarrow BB \mid a \mid \varepsilon$

$B \rightarrow AcA \mid b$

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

1: Zkonstruuuj $N_\varepsilon = \{A \in N \mid A \Rightarrow^* \varepsilon\}$

2: Množinu pravidel P' zkonstruuuj takto:

3: **for all** $A \rightarrow X_1 \dots X_n \in P$ **do**

4: přidej do P' všechna pravidla tvaru $A \rightarrow \alpha_1 \dots \alpha_n$ splňující

5: (a) pokud $X_i \notin N_\varepsilon$ pak $\alpha_i = X_i$

6: (b) pokud $X_i \in N_\varepsilon$ pak α_i je buď X_i nebo ε

7: (c) ne všechna α_i jsou ε

8: **end for**

9: **if** $S \in N_\varepsilon$ **then**

10: přidej do P' pravidla $S' \rightarrow S \mid \varepsilon$

11: $N' := N \cup \{S'\}$

12: **else**

13: $N' := N$; $S' := S$

14: **end if**

Příklad

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

$S \rightarrow aAbBc \mid AB$

$A \rightarrow BB \mid a \mid \varepsilon$

$B \rightarrow AA \mid b$

Korektnost algoritmu

Konečnost.

Výsledná gramatika je bez ε -pravidel.

Ekvivalence gramatik.

Jednoduchá pravidla

Jednoduchým pravidlem nazýváme každé pravidlo tvaru $A \rightarrow B$, kde $A, B \in N$.

$$S \rightarrow aAbBc$$

$$A \rightarrow aA \mid B \mid a$$

$$B \rightarrow bB \mid b$$

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

1: **for all** $A \in N$ **do**

2: $i := 0$; $N_0 := \{A\}$

3: **repeat**

4: $i := i + 1$

5: $N_i := N_{i-1} \cup \{C \mid B \rightarrow C \in P, B \in N_{i-1}\}$

6: **until** $N_i = N_{i-1}$

7: $N_A := N_i$

8: **end for**

9: $P' := \emptyset$

10: **for all** $A \in N$ **do**

11: $P' := P' \cup \{A \rightarrow \alpha \mid B \in N_A \wedge B \rightarrow \alpha \in P \text{ není jednoduché}\}$

12: **end for**

Příklad

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

$S \rightarrow ABC$

$A \rightarrow aA \mid B \mid a$

$B \rightarrow bB \mid A$

$C \rightarrow cC \mid A$

Korektnost algoritmu

Konečnost.

Výsledná gramatika neobsahuje jednoduchá pravidla ani ε -pravidla.

Ekvivalence gramatik:

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

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

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

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

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

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. □