Greibachové normální forma

Dalším kanonickým tvarem, kterým se budeme zabývat, je bezkontextová gramatika v Greibachové normální formě (dále též GNF). Nejdříve si uvedeme definici této formy.

Definice – CFG v Greibachové normální formě

Buď $G = (N, \Sigma, P, S)$ bezkontextová gramatika. Řekněme, že G je v Greibachové normální formě právě tehdy, když je bez ε-pravidel a každé pravidlo z P je tvaru $A \to a\alpha$, kde $A \in N, a \in \Sigma, \alpha \in N^*$.

Poznámka.

Převod do GNF vyžaduje znalost několika operací. Uvedeme si je nejdříve v souhrnu a pak si je popíšeme:

1. substituce nějakého pravidla obsahujícího neterminál A (více viz Lemma o substituci)
2. eliminace bezprostřední levé rekurze – odstranění pravidel $A \to A\alpha$, kde $\alpha \in (N \cup \Sigma)^*$
3. eliminace levé rekurze (neboli nepřímé levé rekurze)
4. algoritmus pro převod do GNF

Poznámka.

Ještě než se začneme zabývat jednotlivými algoritmy, měli bychom si vysvětlit, proč je užitečné převést si nějakou bezkontextovou gramatiku do Greibachové normální formy. Představte si nějaký imaginární nástroj, který bude pracovat dle pravidel gramatiky a číst vstupní slovo. V závislosti na něm bude používat jednotlivá pravidla. Je patrné, že takovému „stroji“ bude platnější, když pravidla budou ve tvaru odpovídajícímu GNF, tj. jejich pravé strany budou začínat terminálem. Bude totiž moci porovnat aktuálně čtený symbol vstupního slova s počátečním terminálem každého pravidla. Zvolí poté to pravidlo, které začíná stejným terminálem jako aktuálně čtený vstupní symbol.

Každopádně, předchozí idea nemusí platit v případě, že gramatika v GNF obsahuje dvě A-pravidla začínající stejným terminálem. V tu chvíli už se stroj opět musí chovat nedeterministicky, tj. volit, jaké pravidlo má použít.

Lemma (o substituci)

Buď $G = (N, \Sigma, P, S)$ bezkontextová gramatika. Mějme pravidlo

$$A \to xBy, \quad B \in N,\ x, y \in (N \cup \Sigma)^*$$

a B-pravidla

$$B \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n, \quad \beta_1, \beta_2, \ldots, \beta_n \in (N \cup \Sigma)^*.$$

Pak gramatika $G_1 = (N, \Sigma, P', S)$, kde

$$P' = P \setminus \{A \to xBy\} \cup \{A \to x\beta_1 y \mid x\beta_2 y \mid \cdots \mid x\beta_n y\},$$

je jazykově ekvivalentní gramatice G.

Příklad 1.

Buď dána bezkontextová gramatika $G = (\{S, A, B\}, \{a, b\}, P, S)$ s pravidly

$$\begin{aligned} P = \{S &\rightarrow aAb \mid bB \\ A &\rightarrow aaAb \mid Baa \\ B &\rightarrow bBb \mid aSb\} \end{aligned}$$

Uvažujme např. pravidlo $A \rightarrow Baa$. Pokud bychom místo pravé strany Baa dosadili $bBbaa$, $aSbaa$ (tj. nahradili symbol B všemi B-pravidly), tak jazyk generovaný novou gramatikou se nezmění. Tj. pomocí pravidel

$$\begin{aligned} P' = \{S &\rightarrow aAb \mid bB \\ A &\rightarrow aaAb \mid bBbaa \mid aSbaa \\ B &\rightarrow bBb \mid aSb\} \end{aligned}$$

dostaneme stejná slova jako v případě pravidel P.

Definice – rekurzivní neterminál

Neterminál $A \in N$ v gramatice $G = (N, \Sigma, P, S)$ se nazývá rekurzivní, právě když v G existuje derivace tvaru $A \Rightarrow^+ \alpha A \beta$, kde $\alpha, \beta \in (N \cup \Sigma)^*$. Navíc,

- je-li $\alpha = \varepsilon$ (tedy $A \Rightarrow^+ A\beta$), pak A se nazývá levorekurzivní.
- pokud $\beta = \varepsilon$ (tedy $A \Rightarrow^+ \alpha A$), pak A se nazývá pravorekurzivní.

Poznámka.

Jistě vás napadne, že pro převod do GNF je potřeba odstranit levorekurzivní neterminály. K tomu slouží algoritmus pro odstranění levé rekurze, který využívá postupu pro odstranění tzv. přímé (neboli bezprostřední) levé rekurze.

Přímou levou rekurzi představují pravidla, která jsou tvaru $A \rightarrow A\alpha$, kde $\alpha \in (N \cup \Sigma)^*$. Nepřímou levou rekurzí rozumíme derivaci ve více než jednom kroku, která vede k levé rekurzi. Například derivace $A \Rightarrow BaaB \Rightarrow AbaaB$ je příčinou toho, že neterminál A je levorekurzivní, avšak daná rekurze není přímá.

Lemma (o odstranění přímé levé rekurze)

Buď $G = (N, \Sigma, P, S)$ bezkontextová gramatika v níž všechna A-pravidla ($A \in N$) jsou tvaru:

$$(*) \quad A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \cdots \mid A\alpha_n \mid \beta_1 \mid \beta_2 \mid \cdots \mid \beta_k,$$

kde každý řetěz β_i začíná jiným symbolem než A.

Nechť $G' = (N \cup \{A'\}, \Sigma, P', S)$, kde pravidla $(*)$ nahradíme takto:

$$\begin{aligned} A &\rightarrow \beta_1 \mid \beta_2 \mid \cdots \mid \beta_k \mid \beta_1 A' \mid \beta_2 A' \mid \cdots \mid \beta_k A' \\ A' &\rightarrow \alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_n \mid \alpha_1 A' \mid \alpha_2 A' \mid \cdots \mid \alpha_n A' \end{aligned}$$

Pak $L(G) = L(G')$.

Příklad 2.

Eliminujte přímou levou rekurzi z gramatiky $G = (\{E, T, F\}, \{+, *, i, (,)\}, P, E)$, kde množina pravidel

$$\begin{aligned} P = \{E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid i\} \end{aligned}$$

Řešení. Po pozorném prozkoumání pravidel zjistíme, že přímá levá rekurze je u pravidel $E \rightarrow E + T,\ T \rightarrow T * F$.

1. Místo E-pravidel $E \rightarrow E + T \mid T$ dosadíme nová E a E'-pravidla:

$$\begin{aligned} E &\rightarrow T \mid TE' \\ E' &\rightarrow +T \mid +TE' \end{aligned}$$

2. Místo T-pravidel $T \rightarrow T * F \mid F$ dosadíme nová T a T'-pravidla:

$$\begin{aligned} T &\rightarrow F \mid FT' \\ E' &\rightarrow *F \mid *FT' \end{aligned}$$

Výsledná gramatika $G' = (\{E, T, F, E', T'\}, \{+, *, i, (,)\}, P', E)$, kde množina pravidel

$$\begin{aligned} P' = \{E &\rightarrow T \mid TE' \\ E' &\rightarrow +TE' \mid +T \\ T &\rightarrow F \mid FT' \\ T' &\rightarrow *F \mid *FT' \\ F &\rightarrow (E) \mid i\} \end{aligned}$$

neobsahuje přímou levou rekurzi a platí $L(G) = L(G')$.

Poznámka.

Je důležité si uvědomit, že použitý algoritmus funguje tak, že zachovává jazyk obou gramatik, tj. $L(G) = L(G')$. V původní gramatice G jsme mohli mít např. tuto derivaci:

$$(1) \quad E \Rightarrow E + T \Rightarrow E + T + T \Rightarrow T + T + T$$

Tedy: používali jsme pravidlo $E \rightarrow E + T$, které zapřičiňuje, že E je levorekurzivní. Derivaci (1) nahradíme v nové gramatice takto:

$$(2) \quad E \Rightarrow TE' \Rightarrow T + TE' \Rightarrow T + T + T$$

Snad je zřejmě, že pravidlo $E + T$ můžeme používat, kolikrát chceme, avšak poté stejně musíme zvolit jiné nerekurzivní pravidlo ($E \rightarrow T$), abychom směřovali k výsledku, tj. terminálnímu řetězu.

Nová pravidla fungují opačně: nejdříve použijeme nerekurzivní pravidlo ($E \rightarrow TE'$) a poté původní levou rekurzi nahrazujeme odvozením pomocí pravidla ($E' \rightarrow +TE'$), kde E' je pravorekurzivní. Nahrazujeme tedy levou rekurzi za pravou…

Příklad 3.

Eliminujte přímou levou rekurzi z gramatiky $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, kde

$$\begin{aligned} P = \{S &\rightarrow AB \mid \varepsilon, \\ A &\rightarrow abB \mid Aa \mid AbbC \mid aC \mid BC \mid ba \\ B &\rightarrow BC \mid BbA \mid Ab \mid bbb, \\ C &\rightarrow CAb \mid AAA \mid aba\} \end{aligned}$$

Algoritmus (odstranění levé rekurze)

Vstup: Vlastní CFG $G = (N, \Sigma, P, S)$
Výstup: Ekvivalentní gramatika bez levorekurzivních neterminálů

Uspořádej libovolně množinu $N - N = \{A_1, A_2, \ldots, A_n\}$
for $i := 1$ **to** n **do**
  for $j := 1$ **to** $i - 1$ **do**
    foreach pravidlo tvaru $A_i \rightarrow A_j\alpha$ **do**
      použij lemma o substituci a přidej pravidlo $A_i \rightarrow \beta_1\alpha \mid \beta_2\alpha \mid \cdots \mid \beta_k\alpha$
        (kde $A_j \rightarrow \beta_1 \mid \beta_2 \mid \cdots \mid \beta_k$ jsou všechna A_j-pravidla);
      vypusť pravidlo $A_i \rightarrow A_j\alpha$.
    od
  od
  foreach pravidlo tvaru $A_i \rightarrow A_i\alpha$ **do**
    odstraň přímou levou rekurzi
  od
od

Poznámka.

1. V předchozím algoritmu byly použity symboly α, $\beta_1, \ldots, \beta_k$, které značí libovolný řetězec terminálů a neterminálů.
2. Algoritmus je korektní v tom smyslu, že nemění jazyk generovaný původní gramatikou. Vysvětlení je nasnadě: používá pouze lemma o substituci a lemma o odstranění přímé levé rekurze. O těchto dvou postupech víme, že zachovávají jazyk.

Poznámka.

Ještě si algoritmus slovně popíšeme. Skládá se v podstatě ze tří částí:

1. uspořádání neterminálů – není důležité, jakým způsobem neterminály uspořádáme, musíme však toto pořadí v další části algoritmu dodržet.
2. procházení množiny neterminálů od prvního k poslednímu a aplikace následujících dvou kroků:

   2a) lemma o substituci: máme-li množinu neterminálů uspořádanou ($N = \{A_1, A_2, \ldots, A_n\}$), pak tento krok nemusíme dělat pro A_1. U ostatních neterminálů A_i ($2 \leq i \leq n$) postupujeme tak, že sledujeme všechna A_i-pravidla tvaru $A_i \rightarrow A_j\alpha$, kde $j < i$, tj. hledáme pravé strany A_i-pravidel začínající na nějaký neterminál v uspořádání před A_i. Pokud takové pravidlo najdeme, provedeme jeho substituci. Velmi důležité je, že pravé strany nahrazujeme postupně od 1. do $i-1$-tého neterminálu, nemůžeme měnit pořadí substitucí. Neméně podstatný je fakt, že každá nová množina pravidel vzniklá díky substituci již slouží jako vstup pro další substituci.

2b) odstranění přímé levé rekurze: po provedení kroku 2a může neterminál A_i obsahovat pouze přímou levou rekurzi, tj. pravidla tvaru $A_i \to A_i\alpha$. Tento druh rekurze odstraníme pomocí lemmatu o odstranění přímé levé rekurze.

Příklad 4.

Odstraňte levou rekurzi z gramatiky $G = (\{S, X, Y\}, \{a, b, c, d\}, P, S)$, kde množina pravidel

$$\begin{aligned} P = \{S &\to Xc \mid Yd \mid Yb \\ X &\to Xd \mid a \\ Y &\to SaS\} \end{aligned}$$

Řešení. Nejdříve uspořádáme množinu neterminálů. Protože nezáleží na pořadí, ponecháme jej takto: $N = \{S, X, Y\}$. Poté provádíme kroky 2a, 2b postupně pro neterminály S, X, Y:

1. Pro první neterminál S nemusíme dělat krok 2a. Ověříme tedy pouze, zda v S-pravidlech není přítomna přímá levá rekurze. Zjistíme, že nikoliv, proto S-pravidla zůstávají beze změny.

2. Zkoumáme další neterminál X. Zjišťujeme, že ani pro něj není třeba provést krok 2a. Krok 2b je však nutno provést, protože je v pravidlech $X \to Xd$. Vytvoříme tedy nový neterminál X' a původní X-pravidla nahradíme. Nová pravidla vypadají takto:

$$\begin{aligned} P' = \{S &\to Xc \mid Yd \mid Yb \\ X &\to a \mid aX' \\ X' &\to d \mid dX' \\ Y &\to SaS\} \end{aligned}$$

3. Posledním neterminálem v pořadí je Y a jediné pravidlo $Y \to SaS$. S je uspořádán před Y, takže použijeme lemma o substituci. Výsledkem je:

$$\begin{aligned} P' = \{S &\to Xc \mid Yd \mid Yb \\ X &\to a \mid aX' \\ X' &\to d \mid dX' \\ Y &\to XcaS \mid YdaS \mid YbaS\} \end{aligned}$$

Všimněte si nyní nového pravidla $Y \to XcaS$. X je v pořadí před neterminálem Y, provedeme tedy ještě jednu substituci:

$$\begin{aligned} P' = \{S &\to Xc \mid Yd \mid Yb \\ X &\to a \mid aX' \\ X' &\to d \mid dX' \\ Y &\to acaS \mid aX'caS \mid YdaS \mid YbaS\} \end{aligned}$$

Krok 2a je hotov, přistoupíme ke kroku 2b. U pravidel $Y \to \mid YdaS \mid YbaS$ můžeme sledovat přímou levou rekurzi. Vytvoříme nový neterminál Y' a změníme Y-pravidla tak, abychom rekurzi odstranili:

$$\begin{aligned}P' = \{S &\rightarrow Xc \mid Yd \mid Yb\\ X &\rightarrow a \mid aX'\\ X' &\rightarrow d \mid dX'\\ Y &\rightarrow acaS \mid aX'caS \mid acaSY' \mid aX'caSY'\\ Y' &\rightarrow daS \mid baS \mid daSY' \mid baSY'\}\end{aligned}$$

Výsledná gramatika $G' = (\{S, X, X', Y, Y'\}, \{a, b, c, d\}, P', S)$ je bez levé rekurze a platí $L(G) = L(G')$.

Poznámka.

Ještě jednou zdůrazníme, že krok 2a provádíme dle pořadí neterminálů a tak dlouho, dokud můžeme. V předchozím příkladu 4 jste si mohli všimnout, že pro poslední neterminál Y jsme substituci prováděli dvakrát.

Příklad 5.

Eliminujte levou rekurzi u bezkontextové gramatiky $G = (\{S, A, B\}, \{a, b\}, P, S)$, kde množina pravidel

$$\begin{aligned}P = \{S &\rightarrow Aa \mid Bb \mid aaA \mid SaA \mid SbB\\ A &\rightarrow AAb \mid ab \mid SBb\\ B &\rightarrow Bbb \mid BBB \mid bAb\}\end{aligned}$$

Algoritmus (převod do GNF)

Vstup: vlastní CFG $G = (N, \Sigma, P, S)$ bez levé rekurze.
Výstup: CFG G' v GNF taková, že $L(G) = L(G')$.

Uspořádej libovolné dva neterminály A, B množiny N tak, že $A << B$, právě když existuje pravidlo $A \rightarrow B\alpha$, kde $\alpha \in (N \cup \Sigma)^*$. Nechť tedy množina $N = \{A_1, A_2, \ldots, A_n\}$ je uspořádaná dle $<<$.

for $i := n-1$ **downto** 1 **do**
  foreach pravidlo tvaru $A_i \rightarrow A_j\alpha$ $(j > i)$
    použij lemma o substituci a přidej pravidlo $A_i \rightarrow \beta_1\alpha, \ldots, \beta_k\alpha$,
      kde $A_j \rightarrow \beta_1 \mid \cdots \mid \beta_k$ jsou všechna A_j-pravidla;
    vypusť pravidlo $A_i \rightarrow A_j\alpha$
  od
od

Pro každé pravidlo nahraď libovolný terminál $a \in \Sigma$ na 2. až poslední pozici novým neterminálem a' a přidej pravidlo $a' \rightarrow a$.

Poznámka.

Předchozí algoritmus si opět slovně popíšeme. Můžeme si jej pracovně rozdělit do tří částí, přičemž ta prostřední bude pravděpodobně nejnáročnější.

1. uspořádání neterminálů – při aplikaci algoritmu pro převod do GNF již neterminály neřadíme libovolně, avšak podle určitého pravidla. Je doporučeno, abyste si prošli celou množinu pravidel a kdykoliv naleznete pravidlo tvaru $A \to B\alpha$, kde $A, B \in N,\ \alpha \in (N \cup \Sigma)^*$, tak si zapište nerovnost $A << B$. Jakmile budete mít souhrn všech nerovností $<<$, můžete přistoupit k vytvoření uspořádání.

2. aplikace lemmatu o substituci – procházíme neterminály v opačném pořadí, tj. od posledního n-tého k prvnímu. Začínáme $(n-1)$-tým neterminálem v pořadí a hledáme pravidlo tvaru $A_{n-1} \to A_n\alpha$. Pokud jej nalezneme, uplatníme lemma o substituci a místo pravé strany $A_n\alpha$ vložíme $\beta_1\alpha \mid \cdots \mid \beta_k\alpha$, kde $A_n \to \beta_1 \mid \cdots \mid \beta_k$ jsou všechna A_n-pravidla. Podobným způsobem postupujeme dále. Je potřeba mít neustále na paměti, že každou substituci pro i-tý neterminál je potřeba uvažovat pro neterminály s nižším číslem. Tj. novou množinu pravidel vždy zahrneme do dalších operací!

3. „očárkování“ – výsledná gramatika může obsahovat pravidla, která mají na 2. až poslední pozici pravé strany libovolný terminál $x \in \Sigma$. To nevyhovuje definici Greibachové normální formy. Pravidla tedy upravíme tak, že každý výskyt x nahradíme novým neterminálem x' a přidáme pravidlo $x' \to x$.

Příklad 6.

Převeďte gramatiku $G = (\{A, B, C, D\}, \{a, b\}, P, A)$ do Greibachové normální formy.

$$\begin{aligned} P = \{A &\to BC, \\ B &\to CD \mid AB, \\ C &\to Aa \mid b, \\ D &\to ba \mid DD\} \end{aligned}$$

Řešení. Pozorný čtenář si zajisté všimne, že gramatika G obsahuje levou rekurzi. Např. pravidlo $D \to DD$ je ukázkou přímé levé rekurze. Převodem na gramatiku bez levé rekurze dostaneme následující gramatiku $G_1 = (\{A, B, C, D\}, \{a, b\}, P_1, A)$, kde

$$\begin{aligned} P_1 = \{A &\to BC \\ B &\to CD \mid CDB' \\ B' &\to CB \mid CBB' \\ C &\to b \mid bC' \\ C' &\to DCa \mid DB'Ca \mid DCaC' \mid DB'CaC' \\ D &\to ba \mid baD' \\ D' &\to D \mid DD'\} \end{aligned}$$

Nyní nás čekají postupně 3 úkoly: uspořádání, poté substituce a na konci očárkování.

1. Postupným procházením pravidel P_1 zjistíme následující nerovnosti:
$A << B,\ B << C,\ B' << C,\ C' << D,\ D' << D$. Všimněte si nyní, že neterminály C, D jsou v nerovnostech vždy napravo, tudíž by měly být největší. Následné uspořádání můžeme provést například takto: $N = \{A, B, B', C', D', C, D\}$.

2. Nyní postupujeme od neterminálu D až k neterminálu A. Neterminál D ponecháme beze změny. Taktéž C není potřeba upravovat, obsahuje pravidla začínající terminály. Dalším v

pořadí je D'. U něj již budeme muset použít substituci, protože pro pravidla $D' \rightarrow D \mid DD'$ platí, že $D' << D$. Zapišme tedy D'-pravidla znovu:

$$D' \rightarrow ba \mid baD' \mid baD'D'$$

Přecházíme k neterminálu C'. I zde musíme použít substituci, a to hned u všech 4 C'-pravidel:

$$C' \rightarrow baCa \mid baB'Ca \mid baCaC' \mid baB'CaC' \mid baD'Ca \mid baD'B'Ca \mid baD'CaC' \mid baD'B'CaC'$$

Nyní neterminál B' a jeho pravidla $B' \rightarrow CB \mid CBB'$. Z uspořádání je patrné, že $B' << C$, použijeme tedy opět substituci:

$$B' \rightarrow bB \mid bC'B \mid bBB' \mid bC'BB'$$

Pokračujeme neterminálem B a je to obdobné jako u B'. Pravidla $B \rightarrow CD \mid CDB'$ nahradíme:

$$B \rightarrow bD \mid bC'D \mid bDB' \mid bC'DB'$$

Posledním neterminál, který máme prozkoumat, je A. Jeho jediné pravidlo $A \rightarrow BC$ splňuje nerovnost $A << B$, tj. musíme jej nahradit. Za neterminál B však dosadíme postupně 4 nové pravé strany B-pravidel!

$$A \rightarrow bDC \mid bC'DC \mid bDB'C \mid bC'DB'C$$

3. Třetím krokem je očárkování, které provedeme a rovnou podáme výslednou množinu pravidel P':

$$\begin{array}{rcl}
A & \rightarrow & bDC \mid bC'DC \mid bDB'C \mid bC'DB'C \\
B & \rightarrow & bD \mid bC'D \mid bDB' \mid bC'DB' \\
B' & \rightarrow & bB \mid bC'B \mid bBB' \mid bC'BB' \\
C' & \rightarrow & ba'Ca' \mid ba'B'Ca' \mid ba'Ca'C' \mid ba'B'Ca'C' \mid ba'D'Ca' \mid ba'D'B'Ca' \mid ba'D'Ca'C' \mid ba'D'B'Ca'C' \\
D' & \rightarrow & ba' \mid ba'D' \mid ba'D'D' \\
C & \rightarrow & b \mid bC' \\
D & \rightarrow & ba' \mid ba'D'
\end{array}$$

Příklad 7.

Převeďte bezkontextovou gramatiku $G = (\{S, X, Y\}, \{5, 6\}, P, S)$ do Greibachové normální formy.

$$\begin{array}{rcl}
P = \{S & \rightarrow & 55 \mid X66 \\
X & \rightarrow & 56 \mid S6 \\
Y & \rightarrow & 65 \mid S5Y\}
\end{array}$$

Věta.

K libovolné bezkontextové gramatice existuje jazykově ekvivalentní CFG v Greibachové normální formě.