Redukované gramatiky

Následující studijní materiál se zabývá tzv. **redukovanými gramatikami**, neboli bezkontextovými gramatikami bez **nepoužitelných symbolů**. Často se nám může stát, že budeme uvažovat bezkontextovou gramatiku, kde je spousta informací redundantních, tj. takových, že neovlivní výsledný jazyk generovaný gramatikou. Můžeme si podat následující příklad gramatiky, která tuto redundanci obsahuje.

Příklad 1.

Je dána bezkontextová gramatika $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, kde množina pravidel

$$
\begin{aligned}
P = \{S &\rightarrow aAb \mid bC \\
A &\rightarrow aAA \\
B &\rightarrow aB \mid bbB \mid b \\
C &\rightarrow bC \mid b
\end{aligned}
$$

Všimněte si neterminálů A, B. Abychom mohli uplatnit pravidla pro neterminál A, tj. abychom při derivaci použili tento symbol, musíme na začátku použít odvození $S \Rightarrow aAb$. Pro další odvozování už můžeme aplikovat pouze pravidlo $A \rightarrow aAA$. To znamená, že výsledkem derivace $S \Rightarrow aAb \Rightarrow aaAAb \Rightarrow \ldots$ nikdy nebude slovo, čili řetězec pouze terminálních symbolů. Takový neterminál je pro nás nadbytečný, nemusíme ho v gramatice mít, protože pomocí něj stejně nic nevygenerujeme.

S neterminálem B je to podobné. Dokonce neexistuje derivace, která by začínala v počátečním neterminálu S a „dosáhla“ neterminálu B. Díky tomu je pro nás neterminál B také zbytečný a můžeme jej z gramatiky vypustit.

Pokud bychom tedy odstranili všechna A-pravidla, B-pravidla a pravidlo $S \rightarrow aAb$, zbyde nám pouze obyčejná regulární gramatika s pravidly

$$
\begin{aligned}
P = \{S &\rightarrow bC \\
C &\rightarrow bC \mid b
\end{aligned}
$$

generující jazyk $L = \{b^i \mid i > 1\}$. Co je však důležité, odstranění neterminálů A, B bylo užitečné, neboť z původní složitější gramatiky zbyla jednodušší gramatika, u které si můžeme lépe představit, co generuje.

Definice – nepoužitelný symbol

Buď $G = (N, \Sigma, P, S)$ bezkontextová gramatika. Symbol $X \in N \cup \Sigma$ nazveme **nepoužitelným**, právě když v G **neexistuje** derivace tvaru

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

pro žádné $w, x, y \in \Sigma^*$.

Definice – redukovaná gramatika

Bezkontextová gramatika $G = (N, \Sigma, P, S)$ se nazývá **redukovaná**, právě když neobsahuje nepoužitelné symboly.

Poznámka.

Nepoužitelnost symbolů si můžeme rozdělit na dva typy:

1. **Nepoužitelnost 1. typu** (nenormovanost): neexistuje $w \in \Sigma^*$ takové, že $A \Rightarrow^* w$.
2. **Nepoužitelnost 2. typu** (nedosažitelnost): neexistují $x, y \in (N \cup \Sigma)^*$ takové, že $S \Rightarrow^* xAy$.

V následujícím textu se budeme zabývat tím, že nepoužitelné symboly 1. a 2. typu odstranit, a také, v jakém pořadí použít dané dva algoritmy tak, aby výsledná gramatika byla redukovaná.

Algoritmus (nalezení nepoužitelných symbolů 1. typu)

Následující algoritmus hledá takové neterminály A, pro něž existuje $w \in \Sigma^*$ tak, že $A \Rightarrow^* w$. Takové neterminály přidává do množiny N_i. Poslední takovou množinu N_i, která už není rozdílná od té předcházející N_{i-1}, přiřadí do výsledné množiny N_e obsahující pouze použitelné symboly 1. typu (tj. normované).

Vstup: CFG $G = (N, \Sigma, P, S)$
Výstup: Množina $N_e = \{A \in N \mid \exists w \in \Sigma^*.\ A \Rightarrow^* w\}$.

$i := 0;\ N_0 := \emptyset$
repeat $i := i + 1$
$\quad N_i := N_{i-1} \cup \{A \in N \mid A \to \alpha \in P, \alpha \in (N_{i-1} \cup \Sigma)^*\}$
until $N_i = N_{i-1}$
$N_e := N_i$

Poznámka.

Algoritmus si ještě raději popíšeme slovně. Kroky probíhají v tomto pořadí:

1. **Počáteční inicializace:** Vytvoříme množinu N_0, která nebude mít žádný prvek.
2. Konstruujeme množinu N_1. Přidáme do ní pouze ty neterminály, z nichž můžeme přímo odvodit terminální řetěz.
3. Konstruujeme množiny $N_i,\ i \in \{2, 3, \dots\}$. Do N_i nejdříve přidáme všechny neterminály patřící do N_{i-1}. Poté uvažujeme ty neterminály A chybějící v množině N_{i-1}. Sledujeme všechna A-pravidla a obsahuje-li některé z nich pouze řetězec terminálů či neterminálů patřících do N_{i-1}, přidáme A do N_i.
4. Krok 3. opakujeme tak dlouho, dokud N_i obsahuje více prvků než N_{i-1}.
5. Položíme $N_e = N_i$.

Poznámka.

Výsledná gramatika neobsahující nepoužitelné symboly 1. typu se získá tak, že místo původní množiny N uvažujeme pouze množinu N_e, přičemž z množiny pravidel P odstraníme všechny pravé strany obsahující neterminály nepatřící do N_e, tj. ta pravidla obsahující nepoužitelné symboly 1. typu.

Příklad 2.

Odstraňte nepoužitelné symboly 1. typu z gramatiky $G = (\{S, A, B, C, D, E, F\}, \{a, b, c\}, P, S)$, kde

$$
\begin{aligned}
P = \{S &\rightarrow aA \mid bB \\
A &\rightarrow aAB \mid aa \mid AC \mid AE \\
B &\rightarrow bBA \mid bb \mid CB \mid BF \\
C &\rightarrow DE \\
D &\rightarrow cc \mid DD \\
E &\rightarrow FF \mid FE \\
F &\rightarrow EcE\}
\end{aligned}
$$

Řešení.

1. Nejdříve provedeme počáteční inicializaci množin N_i, tj. nastavíme $N_0 = \emptyset$.

2. Do množiny N_1 přidáme ty neterminály, které se dají přímo derivovat na terminální řetěz, tedy symboly A (díky pravidlu $A \rightarrow aa$), B (díky pravidlu $B \rightarrow bb$) a D (díky pravidlu $D \rightarrow cc$). Platí tedy, že $N_1 = \{A, B, D\}$.

3. Uvažujeme, co přidat do množiny N_2 kromě neterminálů A, B, D, které do ní už automaticky patří. Např. neterminál S obsahuje pravidlo $S \rightarrow aA$ – pravá strana se skládá z terminálu a a neterminálu $A \in N_1$. S můžeme do N_2 přidat. Totéž ale neplatí pro zbylé neterminály C, E, F. Množina $N_2 = \{S, A, B, D\}$.

4. Uvažujme zbylé neterminály. Neterminálu C se týká jediné pravidlo $C \rightarrow DE$, kde sice $D \in N_2$, avšak $E \notin N_2$. To znamená, že C nemůžeme přidat do N_3. Sledujme dále obě E-pravidla. Pravidlo $E \rightarrow FF$ nám nepomůže – vždyť $F \notin N_2$. Stejně tak pravidlo $E \rightarrow FE$, ani neterminál E nemůžeme přidat do N_3. Posledním uvažovaným neterminálem je F, který bohužel také nemůžeme zahrnout do N_3, neboť pravidlo $F \rightarrow EcE$ obsahuje neterminál $E \notin N_2$.

5. Jelikož $N_3 = N_2$, hledaná množina $N_e = \{S, A, B, D\}$.

Výsledná gramatika bez nepoužitelných symbolů 1. typu je následující: $G' = (\{S, A, B, D\}, \{a, b, c\}, P', S)$, kde

$$
\begin{aligned}
P' = \{S &\rightarrow aA \mid bB \\
A &\rightarrow aAB \mid aa \\
B &\rightarrow bBA \mid bb \\
D &\rightarrow cc \mid DD\}
\end{aligned}
$$

Příklad 3.

Je dána bezkontextová gramatika $G = (\{S, A, B, C, D, E\}, \{a, b, c\}, P, S)$, kde

$$\begin{aligned} P = \{S &\rightarrow aAB \mid bBC \mid cE \\ A &\rightarrow aA \mid bD \mid cC \\ B &\rightarrow aBS \mid ab \\ C &\rightarrow DE \\ D &\rightarrow cB \mid DD \\ E &\rightarrow cEE\} \end{aligned}$$

Nalezněte ekvivalentní bezkontextovou gramatiku bez nepoužitelných symbolů 1. typu.

Poznámka.

Uvedený algoritmus se po mírném doplnění používá též ke zjištění toho faktu, zda daná gramatika G generuje prázdný či neprázdný jazyk. Uvědomte si, že pokud startovní neterminál S nepatří do množiny N_e, nelze z něj odvodit žádný terminální řetěz. To však znamená, že $L(G) = \emptyset$. Pokud $S \in N_e$, tak je jazyk $L(G)$ neprázdný.

Algoritmus (nalezení nepoužitelných symbolů 2. typu)

2. algoritmus, který probereme, odstraňuje tzv. nedosažitelné symboly, neboli takové symboly A, které nelze odvodit z počátečního neterminálu S. Přesněji to znamená, že neexistují $x, y \in (N \cup \Sigma)^*$ takové, že $S \Rightarrow^* xAy$.

Vstup: CFG $G = (N, \Sigma, P, S)$
Výstup: CFG $G' = (N', \Sigma', P', S)$ bez nepoužitelných symbolů 2. typu a splňující $L(G) = L(G')$.

$i := 0;\ V_0 := \{S\}$
repeat $i := i + 1$
    $V_i := V_{i-1} \cup \{X \in N \cup \Sigma \mid \exists A \in V_{i-1}.\ (A \rightarrow \alpha X \beta \in P)\}$
until $V_i = V_{i-1}$
$N' := N \cap V_i$
$\Sigma' := \Sigma \cap V_i$
$P' := P \cap (V_i \times V_i^*)$

Poznámka.

Raději si předchozí algoritmus opět slovně popíšeme. Naznačíme si v jednotlivých krocích, jak by měl „uživatel“ algoritmu postupovat.

1. **Počáteční inicializace:** Do množiny V_0 vložíme pouze neterminál S.

2. Do množiny V_1 přidáme S (tj. jediný prvek množiny V_0) a dále všechny terminální i neterminální symboly vyskytující se na pravé straně S-pravidel.

3. V dalších krocích konstruujeme množiny V_i, $i \in \{2, 3, \dots\}$. Provádíme to tím způsobem, že vybíráme nové neterminály A, které ještě nebyly zahrnuty ve V_{i-1}. Do množiny V_i vkládáme ty symboly (terminální či neterminální), které se vyskytují na pravé straně všech A-pravidel a nejsou prvkem V_{i-1}.
4. Pokračujeme tak dlouho, dokud V_i je různé od V_{i-1}.
5. Posledním krokem je úprava původní gramatiky G. Do nové gramatiky G' zahrneme pouze ty terminály a neterminály patřící do V_i a samozřejmě odstraníme pravidla obsahující symboly nepatřící do V_i, tj. nepoužitelné symboly 2. typu.

Příklad 4.

Je dána následující gramatika $G = (\{S, A, B, C, D\}, \{a, b, c\}, P, S)$, kde

$$
\begin{aligned}
P = \{S &\rightarrow aA \mid bB \\
A &\rightarrow aAB \mid aa \\
B &\rightarrow bBA \mid bb \\
C &\rightarrow cD \mid bCa \\
D &\rightarrow cc \mid DD\}
\end{aligned}
$$

Převeďte gramatiku G na ekvivalentní gramatiku G' bez nepoužitelných symbolů 2. typu.

Řešení.

1. Opět nejdříve provedeme počáteční inicializaci množiny V_i, tj. položíme $V_0 = \{S\}$.
2. Sledujeme všechna S-pravidla a do V_1 přidáme symboly vyskytující jejich pravé straně. Díky pravidlu $S \rightarrow aA$ přidáme a, A, díky pravidlu $S \rightarrow bB$ vložíme b, B. Množina $V_1 = \{S, A, B, a, b\}$ – nesmíme zapomenout na S, které bylo v předchozí množině V_0.
3. V množině V_1 se nově objevily neterminály A, B. Pokud však projdeme všechna A-pravidla a B-pravidla, tak nic nového do V_2 nepřibývá, tudíž $V_2 = V_1$.
4. Nová gramatika G' bude obsahovat pouze symboly množiny V_2, proto z ní odstraníme symboly C, D, c a všechna C-pravidla a D-pravidla.

Výsledkem bude $G' = (\{S, A, B\}, \{a, b\}, P', S)$, kde

$$
\begin{aligned}
P = \{S &\rightarrow aA \mid bB \\
A &\rightarrow aAB \mid aa \\
B &\rightarrow bBA \mid bb\}
\end{aligned}
$$

Příklad 5.

Je dána následující gramatika $G = (\{S, A, B, C, D, E\}, \{a, b, c, d\}, P, S)$, kde

$$
\begin{aligned}
P = \{S &\rightarrow aBC \\
A &\rightarrow cAD \mid ccA \\
B &\rightarrow bBB \mid EE \\
C &\rightarrow Cd \mid CCa \\
D &\rightarrow AcD \\
E &\rightarrow ECd \mid aBC\}
\end{aligned}
$$

Převeďte gramatiku G na ekvivalentní gramatiku G' bez nepoužitelných symbolů 2. typu.

Algoritmus (převod CFG na redukovanou gramatiku)

Poslední, 3. algoritmus je nejjednodušší a využívá předchozích dvou. Je však velmi důležité, abyste zachovali správné pořadí použití obou algoritmů.

Vstup: CFG $G = (N, \Sigma, P, S)$
Výstup: Redukovaná gramatika G_2 taková, že $L(G) = L(G_2)$.

1. Použijte 1. algoritmus pro odstranění nepoužitelných symbolů 1. typu – výsledkem je gramatika G_1 taková, že $L(G_1) = L(G)$.
2. Použijte 2. algoritmus pro odstranění nepoužitelných symbolů 2. typu – aplikujte jej na gramatiku G_1. Výsledkem je redukovaná gramatika G_2 taková, že $L(G_2) = L(G_1) = L(G)$.

Příklad 6.

K demonstraci faktu, že nelze měnit pořadí použití jednotlivých algoritmů, předkládáme následující příklad gramatiky $G = (\{S, A, B\}, \{a, b\}, P, S)$, kde množina pravidel

$$\begin{array}{rcl} P = \{S & \to & a \mid A \\ A & \to & AB \\ B & \to & b\} \end{array}$$

Použijeme-li správné pořadí, tak nejdříve odstraňujeme nenormované symboly, tj. takové, které se nedají odvodit na terminální řetěz. Postupně přidáváme do množiny N_e neterminály S, B. Neterminál A, jeho pravidla i všechny pravé strany, kde se vyskytuje, musíme odstranit. Výsledkem je gramatika

$$G_1 = (\{S, B\}, \{a, b\}, \{S \to a, B \to b\}, S).$$

Aplikací 2. algoritmu odstraňujeme nedosažitelné symboly, tj. takové, které se nedají odvodit z počátečního neterminálu S. Zjistíme, že ani neterminál B, ani terminál b není dosažitelný z S. Zůstává nám tedy výsledná gramatika

$$G_2 = (\{S\}, \{a\}, \{S \to a\}, S).$$

Pokud bychom však použili opačné pořadí a začali 2. algoritmem, zjistíme postupně, že jsou dosažitelné všechny symboly gramatiky G. Následnou aplikací 1. algoritmu na stejnou gramatiku G odstraňujeme nenormované symboly, tj. takové, které nelze vyderivovat na terminální řetěz. Je zřejmé, že neterminál A opět nezahrneme do výsledné gramatiky, avšak neterminál B ano, což je chyba. Všimněte si, že následující derivací

$$S \Rightarrow^* A \Rightarrow^* AB \Rightarrow^* Ab \Rightarrow^* \dots$$

se k němu sice dostaneme, ale není nám to nic platné, protože z neterminálu A terminální řetězec nedostaneme.

Příklad 7.

Je dána následující gramatika $G = (\{S, A, B, C, D, E\}, \{a, b, c, d\}, P, S)$, kde

$$
\begin{aligned}
P = \{S &\rightarrow acE \mid bSD \mid dB \\
A &\rightarrow abA \mid cdC \\
B &\rightarrow db \mid dB \\
C &\rightarrow caA \mid bdC \mid ab \\
D &\rightarrow bB \\
E &\rightarrow bdE \mid acDE\}
\end{aligned}
$$

Převeďte gramatiku G na ekvivalentní redukovanou gramatiku G'.

Příklad 8.

Nechť $G = (N, \Sigma, P, S)$ je bezkontextová gramatika. Řekneme, že

- neterminál $X \in N$ je **normovaný**, pokud existuje $w \in \Sigma^*$ takové, že $X \Rightarrow^* w$.
- neterminál $Y \in N$ je **dosažitelný**, pokud existují $\alpha, \beta \in (N \cup \Sigma)^*$ takové, že $S \Rightarrow^* \alpha Y \beta$.

Zkonstruujte bezkontextovou gramatiku obsahující neterminály A, B, C, D, pro které platí

- A je nedosažitelný a normovaný,
- B je dosažitelný a nenormovaný,
- C je nedosažitelný a nenormovaný,
- D je dosažitelný, normovaný a nepoužitelný.

Terminály volte libovolně. Gramatika může obsahovat i další neterminály.