1. Uvažte jazyk $L_1 = \{w_1 \# w_2 \mid w_1, w_2 \in \{a,b\}^* \wedge (\#_a(w_1) = \#_a(w_2) \vee \#_b(w_1) = \#_b(w_2))\}$, kde $\#_x(w)$ je počet symbolů x ve slově w.

   (a) Sestavte gramatiku G_1 takovou, že $L(G_1) = L_1$.

   $G_1 = (\{S, X, Y, A, B, \}, \{a, b, \#\}, P, S)$, kde

   $$\begin{array}{rrcl} P: & S & \to & X \mid Y \\ & X & \to & AXA \mid \# \\ & Y & \to & BYB \mid \# \\ & A & \to & Ab \mid bA \mid a \\ & B & \to & aB \mid Ba \mid b \end{array}$$

   (b) Algoritmickým postupem převeďte gramatiku G_1 na zásobníkový automat provádějící syntaktickou analýzu zdola nahoru.

   - Převedeme BKG G_1 na RZA M algoritmem popsaným ve větě **4.15** ve studijní opoře.
   - $M = (\{q, r\}, \{a, b, \#\}, \{S, X, Y, A, B, \} \cup \{a, b, \#\} \cup \{\$\}, \delta, q, \$, \{r\})$, kde zobrazení δ je definováno takto:

     i. $\delta(q, a, \epsilon) = \{(q, a)\}$ pro všechna $a \in \{a, b, \#\}$:
     $\delta(q, a, \epsilon) = \{(q, a)\}$
     $\delta(q, b, \epsilon) = \{(q, b)\}$
     $\delta(q, \#, \epsilon) = \{(q, \#)\}$

     ii. Je-li $A \to \alpha$ pravidlo z P, pak $\delta(q, \epsilon, \alpha)$ obsahuje (q, A):
     $\delta(q, \epsilon, X) = \{(q, S)\}$
     $\delta(q, \epsilon, Y) = \{(q, S)\}$
     $\delta(q, \epsilon, AXA) = \{(q, X)\}$
     $\delta(q, \epsilon, \#) = \{(q, X), (q, Y)\}$
     $\delta(q, \epsilon, BYB) = \{(q, Y)\}$
     $\delta(q, \epsilon, Ab) = \{(q, A)\}$
     $\delta(q, \epsilon, bA) = \{(q, A)\}$
     $\delta(q, \epsilon, a) = \{(q, A)\}$
     $\delta(q, \epsilon, aB) = \{(q, B)\}$
     $\delta(q, \epsilon, Ba) = \{(q, B)\}$
     $\delta(q, \epsilon, b) = \{(q, B)\}$

     iii. $\delta(q, \epsilon, S\$) = \{(r, \epsilon)\}$

   (c) Lze jazyk L_1 přijmout deterministickým zásobníkovým automatem (DZA)? Zdůvodněte své tvrzení (formální důkaz se nepožaduje).

   Jazyk L_1 nelze přijmout deterministickým zásobníkovým automatem, protože automat nemůže dopředu vědět, jestli má kontrolovat podmínku $(\#_a(w_1) = \#_a(w_2)$ nebo podmínku $(\#_b(w_1) = \#_b(w_2)$. Proto automat přijímající tento jazyk by musel mít dva zásobníky (jeden pro symbol a, druhý pro symbol b) nebo být nedeterministický.

2. Mějme jazyk $L_2 = \{w_1 \# w_2 \mid w_1, w_2 \in \{a,b\}^* \wedge \#_a(w_1) = \#_a(w_2) \wedge \#_b(w_1) = \#_b(w_2)\}$. Dokažte, že L_2 není bezkontextový.

   (a) Předpokládejme, že jazyk L_2 je bezkontextový. Pak podle Pumping lemma pro bezkontextové jazyky platí:
   $\exists k > 0 : \forall z \in L_2 : |z| \geq k \Rightarrow \exists u,v,w,x,y \in \{a,b,\#\}^* : z = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L_2$

   (b) Uvažme libovolné $k > 0$ a zvolme $z = b^ka^k\#b^ka^k$. Zřejmě $z \in L_2$ a $|z| = 4k+1 \geq k$ a tedy:
   $\exists u,v,w,x,y \in \{a,b,\#\}^* : z = b^ka^k\#b^ka^k = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L_2$

   (c) Uvažme libovolné $u,v,w,x,y \in \{a,b,\#\}^*$ takové, že:
   $z = b^ka^k\#b^ka^k = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L_2$

   (d) Zvolme $i = 2 \geq 0$ a tedy:
   $z = b^ka^k\#b^ka^k = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge uv^2wx^2y \in L_2$

   (e) Z toho je zřejmé, že aby platilo, že $uv^2wx^2y \in L_2$, musely by pro řetězec vx současně platit následující podmínky:
   - Řetězec vx by musel obsahovat stejný počet symbolů a z podřetězce $b^ka^k\#$ jako symbolů a z podřetězce $\#b^ka^k$.
   - Řetězec vx by musel obsahovat stejný počet symbolů b z podřetězce $b^ka^k\#$ jako symbolů b z podřetězce $\#b^ka^k$.
   - Řetězec vx by nesměl obsahovat symbol $\#$.

   (f) Z toho a z podmínky $vx \neq \epsilon$ je zřejmé, že pro řetězec vwx by musely současně platit následující podmínky:
   - Řetězec v by musel být neprázdný a obsahovat pouze symboly z podřetězce $b^ka^k\#$.
   - Řetězec w by musel obsahovat symbol $\#$.
   - Řetězec x by musel být neprázdný a obsahovat pouze symboly z podřetězce $\#b^ka^k$.

   (g) Z toho a z podmínky $|vwx| \leq k$ je zřejmé, že:
   - Řetězec v by mohl obsahovat pouze symboly a z podřetězce $b^ka^k\#$.
   - Řetězec x by mohl obsahovat pouze symboly b z podřetězce $\#b^ka^k$.

   (h) Pak by ovšem nemohly platit podmínky z bodu (e), tedy $uv^2wx^2y \notin L_2$, což je SPOR.

   (i) Pro řetězec $z = b^ka^k\#b^ka^k$ tedy nelze nalézt rozdělení na podřetězce u, v, w, x, y tak, aby platilo:
   $z = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge uv^2wx^2y \in L_2$.
   Jazyk L_2 tudíž není bezkontextový.

3. Nechť $G = (N, \Sigma, P, S)$ je bezkontextová gramatika a $A \in N$ neterminál. Navrhněte algoritmus, který určí, zda je možné vygenerovat větnou formu obsahující alespoň dva současné výskyty neterminálu A. Formálně: algoritmus určí, zda-li existuje v gramatice G derivace $S \stackrel{*}{\Rightarrow} \alpha A \beta A \gamma$, kde $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$.

   Nápověda: můžete využít podobný trik jako v důkaze věty 6.7 (část 2) ve slidech.

   Nejprve popíšeme algoritmus pro sestrojení průniku konečného automatu se zásobníkovým automatem.

   Algoritmus 1:

   Vstup: Zásobníkový automat $M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_0^1, Z_0^1, F_1)$
   Konečný automat $M_2 = (Q_2, \Sigma_2, \delta_2, q_0^2, F_2)$

   Výstup: Zásobníkový automat $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ takový, že $L(M) = L(M_1) \cap L(M_2)$

   Metoda:

   - $Q = Q_1 \times Q_2$
   - $\Sigma = \Sigma_1 \cap \Sigma_2$
   - $\Gamma = \Gamma_1$
   - $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$ taková, že:

     (a) $\forall q_1^1, q_2^1 \in Q_1 \ \ \forall q_1^2, q_2^2 \in Q_2 \ \ \forall a \in \Sigma \ \ \forall z \in \Gamma \ \ \forall \gamma \in \Gamma^* :$

     $$((q_2^1, q_2^2), \gamma) \in \delta((q_1^1, q_1^2), a, z) \Leftrightarrow (q_2^1, \gamma) \in \delta_1(q_1^1, a, z) \wedge q_2^2 \in \delta_2(q_1^2, a)$$

     (b) $\forall q_1^1, q_2^1 \in Q_1 \ \ \forall q_1^2, q_2^2 \in Q_2 \ \ \forall z \in \Gamma \ \ \forall \gamma \in \Gamma^* :$

     $$((q_2^1, q_2^2), \gamma) \in \delta((q_1^1, q_1^2), \epsilon, z) \Leftrightarrow (q_2^1, \gamma) \in \delta_1(q_1^1, \epsilon, z) \wedge q_1^2 = q_2^2$$

   - $q_0 = (q_0^1, q_0^2)$
   - $Z_0 = Z_0^1$
   - $F = F_1 \times F_2$

Nyní popíšeme algoritmus, který určí, zda-li existuje v bezkontextové gramatice $G = (N, \Sigma, P, S)$ pro neterminál $A \in N$ derivace $S \overset{*}{\Rightarrow} \alpha A \beta A \gamma$, kde $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$.

Algoritmus 2:

Vstup: Bezkontextová gramatika $G = (N, \Sigma, P, S)$ a neterminál $A \in N$

Výstup: ANO pokud existuje v gramatice G derivace $S \overset{*}{\Rightarrow} \alpha A \beta A \gamma$, kde $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$, NE v opačném případě.

Metoda:

- Sestroj BKG $G' = (N, \Sigma', P', S)$ takto:
  (a) Polož $\Sigma' = \Sigma \cup \{a', n'\}$, kde a', n' jsou nové terminály, $\Sigma \cap \{a', n'\} = \emptyset$.
  (b) Polož $P' = P$.
  (c) Do P' přidej nové přepisovací pravidlo $A \to a'$.
  (d) Pro každé $B \in N \backslash \{A\}$ přidej do P' nové přepisovací pravidlo $B \to n'$.

- Převeď BKG G' na ekvivalentní RZA M_1 postupem uvedeným ve větě **4.15** ve studijní opoře.

- Převeď RZA M_1 na ekvivalentní ZA M_2 postupem uvedeným v důkazu věty **4.11** ve studijní opoře.

- Sestroj KA $M_3 = (Q, \Sigma', \delta, q_0, F)$ takový, že:

  $L(M_3) = \{w \in \Sigma'^* \mid w = \alpha a' \beta a' \gamma \wedge \alpha, \beta, \gamma \in \Sigma'^*\}$ takto:

  $Q = \{q_0, q_1, q_2\}$
  $F = \{q_2\}$

  Pro terminál a' a pro všechna $a \in \Sigma' \backslash \{a'\}$ polož:

  $\delta(q_0, a) = \{q_0\}$
  $\delta(q_0, a') = \{q_1\}$
  $\delta(q_1, a) = \{q_1\}$
  $\delta(q_1, a') = \{q_2\}$
  $\delta(q_2, a) = \{q_2\}$
  $\delta(q_2, a') = \{q_2\}$

- Sestroj ZA M_4 takový, že:
  $L(M_4) = L(M_2) \cap L(M_3)$ s pomocí výše popsaného **algoritmu 1**.

- Převeď ZA M_4 na ekvivalentní BKG G_4 postupem uvedeným v důkazu věty **4.16** ve studijní opoře.

- Pro BKG G_4 proveď algoritmus **4.1** ze studijní opory, který určí, zda $L(G_4) \neq \emptyset$. Výstup algoritmu **4.1** je zároveň výstupem tohoto algoritmu, tedy pokud $L(G_4) \neq \emptyset$, pak je výstup ANO, jinak NE.

4. Uvažujte jazyk L_4, který je generován gramatikou
$G_4 = (\{E, T, F\}, \{(,), true, or, not\}, P, E)$, kde

$$\begin{array}{rrcl} P: & E & \to & E\ or\ T \mid T \\ & T & \to & (E) \mid not(E) \mid true \end{array}$$

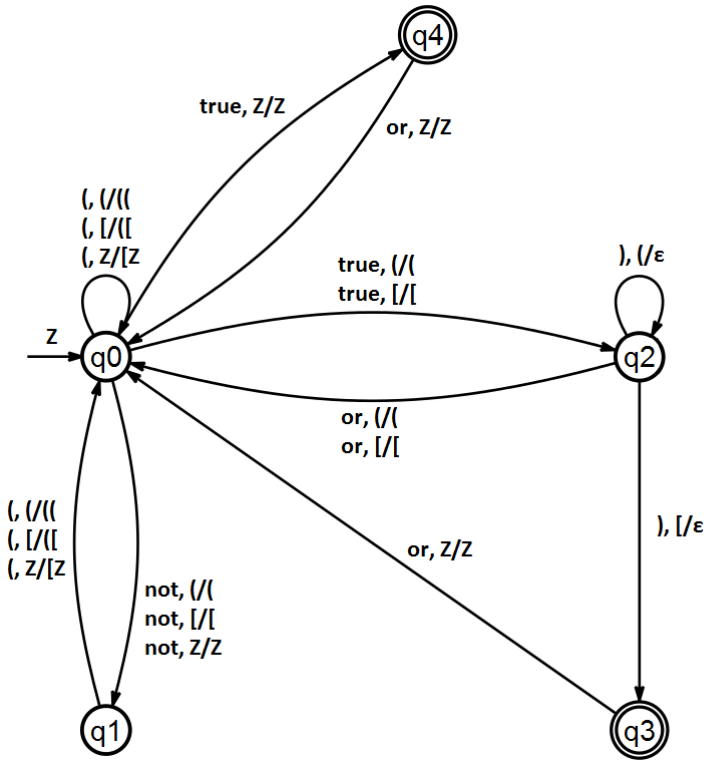
Sestrojte *deterministický* zásobníkový automat přijímající *doplněk* jazyka L_4 a demonstrujte jeho funkci na zamítnutí řetězce $(true\, or\, not(true))\, or\, true$. Vhodným postupem je nejprve vytvořit deterministický ZA přijímající jazyk L_4 a následně vytvořit jeho doplněk.

Poznámka: v souladu s definicí gramatiky jsou $true$, or a not jednotlivými symboly, *nikoliv* řetězci znaků.

Nejprve vytvoříme DZA M_1 přijímající jazyk L_4. $M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_0, Z, F_1)$, kde:

$$\begin{aligned} Q_1 &= \{q_0, q_1, q_2, q_3, q_4\} \\ \Sigma_1 &= \{(,), true, or, not\} \\ \Gamma_1 &= \{Z, (, [\} \\ F_1 &= \{q_3, q_4\} \end{aligned}$$

Přechodovou funkci δ_1 znázorňuje diagram:

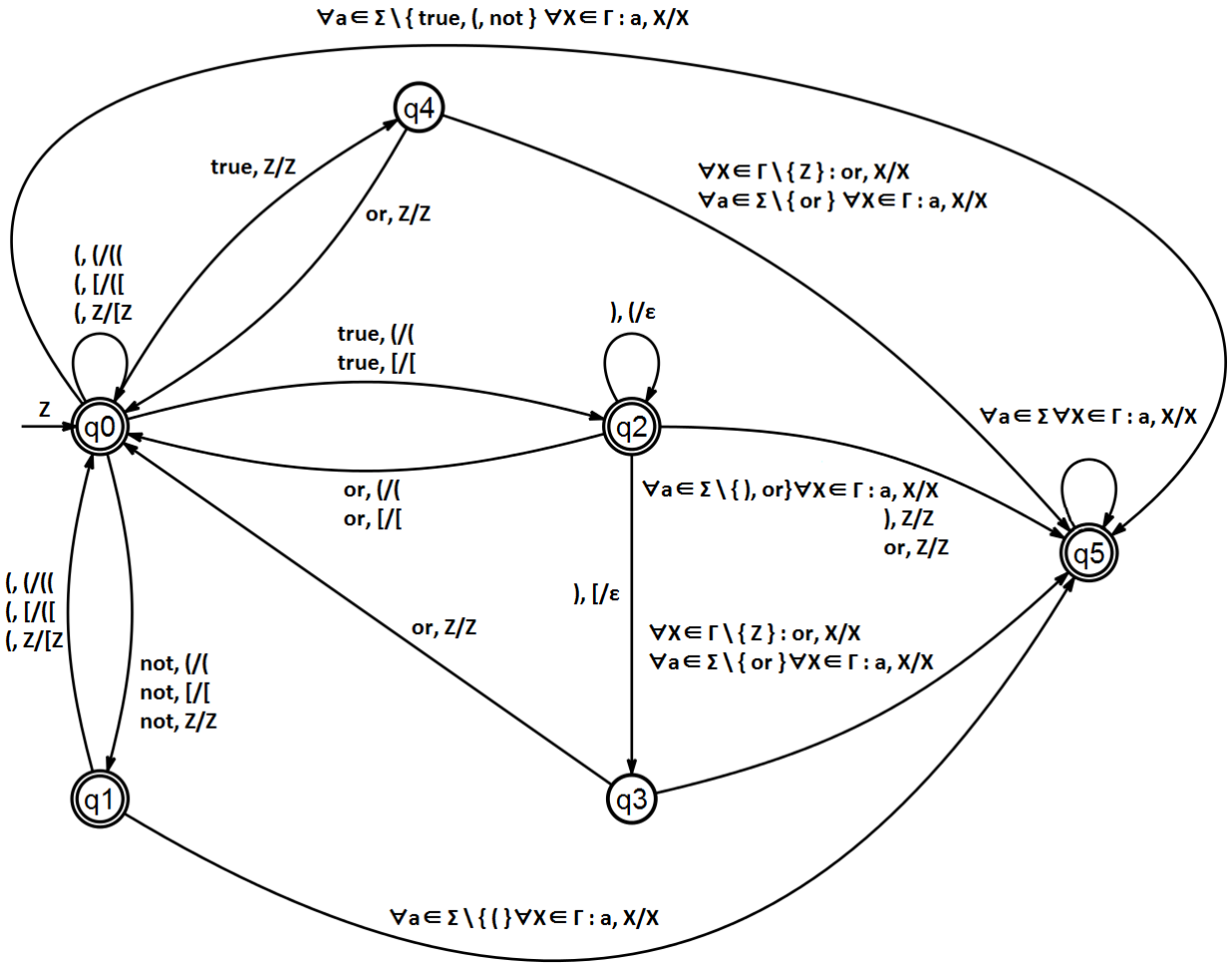


DZA M_2 přijímající doplněk jazyka L_4 získáme z DZA M_1 záměnou koncových a nekoncových stavů a přidáním sink stavu.

$M_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, q_0, Z, F_2)$, kde:

$$\begin{aligned} Q_2 &= \{q_0, q_1, q_2, q_3, q_4, q_5\} \\ \Sigma_2 &= \{(,), true, or, not\} \\ \Gamma_2 &= \{Z, (, [\} \\ F_2 &= \{q_0, q_1, q_2, q_5\} \end{aligned}$$

Přechodovou funkci δ_2 znázorňuje diagram:



Pro vstupní řezězec $(true\ or\ not(true))\ or\ true$ realizuje automat M_2 tyto přechody:

$$
\begin{aligned}
(q_0, (true\ or\ not(true))\ or\ true, Z) &\vdash (q_0, true\ or\ not(true))\ or\ true, [Z) \\
&\vdash (q_2, or\ not(true))\ or\ true, [Z) \\
&\vdash (q_0, not(true))\ or\ true, [Z) \\
&\vdash (q_1, (true))\ or\ true, [Z) \\
&\vdash (q_0, true))\ or\ true, ([Z) \\
&\vdash (q_2,))\ or\ true, ([Z) \\
&\vdash (q_2,)\ or\ true, [Z) \\
&\vdash (q_3, or\ true, Z) \\
&\vdash (q_0, true, Z) \\
&\vdash (q_4, \epsilon, Z)
\end{aligned}
$$

Automat se po přečtení celého řetězce nachází v nekoncovém stavu, tento řetězec tedy nebyl automatem přijat.

5. Mějme gramatiku $G_5 = (\{S, A, B\}, \{a, b, c\}, P, S)$, kde

$$\begin{aligned} P: \quad & S \rightarrow aABa \\ & A \rightarrow B \mid a \mid c \\ & B \rightarrow Bc \mid bB \mid \epsilon \end{aligned}$$

Převeďte gramatiku G_5 *algoritmicky* do (a) Chomského normální formy a (b) Greibachové normální formy.

Nejprve převedeme gramatiku G_5 na ekvivalentní vlastní gramatiku odstraněním ϵ-pravidel, zbytečných symbolů a cyklů, přičemž cykly odstraníme odstraněním ϵ-pravidel a jednoduchých pravidel:

- Převod gramatiky G_5 na ekvivalentní gramatiku bez ϵ-pravidel G_6 s pomocí algoritmu **4.4** ze studijní opory:

  Krok (1):

  $N_0 = \emptyset$
  $N_1 = \{A \mid A \rightarrow \alpha \in P \wedge \alpha \in (N_0 \cup \epsilon)^*\} \cup N_0 = \{B\}$
  $N_2 = \{A \mid A \rightarrow \alpha \in P \wedge \alpha \in (N_1 \cup \epsilon)^*\} \cup N_1 = \{A, B\}$
  $N_3 = \{A \mid A \rightarrow \alpha \in P \wedge \alpha \in (N_2 \cup \epsilon)^*\} \cup N_2 = \{A, B\} = N_2 = N_\epsilon$

  Krok (2a):

  Do nové množiny přepisovacích pravidel P_6 přidáme:

$$\begin{aligned} S &\rightarrow aABa \mid aBa \mid aAa \mid aa \\ A &\rightarrow B \mid a \mid c \\ B &\rightarrow Bc \mid bB \mid c \mid b \end{aligned}$$

  Krok (2b):

  Protože $S \notin N_\epsilon$, množina neterminálů i počáteční symbol gramatiky G_6 jsou stejné jako u gramatiky G_5.

  Krok (3):

  Výsledná gramatika bez ϵ-pravidel:

  $G_6 = (\{S, A, B\}, \{a, b, c\}, P_6, S)$, kde:

$$\begin{aligned} P_6: \quad & S \rightarrow aABa \mid aBa \mid aAa \mid aa \\ & A \rightarrow B \mid a \mid c \\ & B \rightarrow Bc \mid bB \mid c \mid b \end{aligned}$$

- Převod gramatiky bez ϵ-pravidel G_6 na ekvivalentní gramatiku bez ϵ-pravidel a bez jednoduchých pravidel G_7 s pomocí algoritmu **4.5** ze studijní opory:

  Krok (1):

  $N_0 = \{S\}$
  $N_1 = \{C \mid B \to C \in P_6 \wedge B \in N_0\} \cup N_0 = \{S\}$
  $N_1 = N_0, N_S = N_1 = \{S\}$

  $N_0 = \{A\}$
  $N_1 = \{C \mid B \to C \in P_6 \wedge B \in N_0\} \cup N_0 = \{A, B\}$
  $N_1 \neq N_0$
  $N_2 = \{C \mid B \to C \in P_6 \wedge B \in N_1\} \cup N_1 = \{A, B\}$
  $N_2 = N_1, N_A = N_2 = \{A, B\}$

  $N_0 = \{B\}$
  $N_1 = \{C \mid B \to C \in P_6 \wedge B \in N_0\} \cup N_0 = \{B\}$
  $N_1 = N_0, N_B = N_1 = \{B\}$

  Krok (2):

  $$\begin{array}{lrcl} P_7: & S & \to & aABa \mid aBa \mid aAa \mid aa \\ & A & \to & Bc \mid bB \mid a \mid c \mid b \\ & B & \to & Bc \mid bB \mid c \mid b \end{array}$$

  Krok (3):

  Výsledná gramatika bez ϵ-pravidel a jednoduchých pravidel:

  $G_7 = (\{S, A, B\}, \{a, b, c\}, P_7, S)$.

- Převod gramatiky bez ϵ-pravidel a jednoduchých pravidel G_7 na ekvivalentní gramatiku bez ϵ-pravidel, jednoduchých pravidel a zbytečných symbolů (a tedy vlastní gramatiku) G_8 s pomocí algoritmu **4.3** ze studijní opory:

  Krok (1):

  $N_0 = \emptyset$
  $N_1 = \{A \mid A \to \alpha \in P_7 \wedge \alpha \in (N_0 \cup \Sigma)^*\} \cup N_0 = \{S, A, B\}$
  $N_2 = \{A \mid A \to \alpha \in P_7 \wedge \alpha \in (N_1 \cup \Sigma)^*\} \cup N_1 = \{S, A, B\} = N_1 = N_t$

  $\overline{G_7} = (\{S, A, B\}, \{a, b, c\}, \overline{P_7}, S)$, kde:

  $$\begin{array}{lrcl} \overline{P_7}: & S & \to & aABa \mid aBa \mid aAa \mid aa \\ & A & \to & Bc \mid bB \mid a \mid c \mid b \\ & B & \to & Bc \mid bB \mid c \mid b \end{array}$$

  Krok (2):

  $V_0 = \{S\}$
  $V_1 = \{X \mid A \to \alpha X \beta \in \overline{P_7} \wedge A \in V_0\} \cup V_0 = \{S, A, B, a\}$
  $V_2 = \{X \mid A \to \alpha X \beta \in \overline{P_7} \wedge A \in V_1\} \cup V_1 = \{S, A, B, a, b, c\}$
  $V_3 = \{X \mid A \to \alpha X \beta \in \overline{P_7} \wedge A \in V_2\} \cup V_2 = \{S, A, B, a, b, c\} = V_2$

  Výsledná vlastní gramatika:

  $G_8 = (N_8, \Sigma_8, P_8, S)$, kde:

  $$\begin{array}{rcl} N_8 & = & V_3 \cap \{S, A, B\} = \{S, A, B\} \\ \Sigma_8 & = & V_3 \cap \{a, b, c\} = \{a, b, c\} \end{array}$$

  $$\begin{array}{lrcl} P_8: & S & \to & aABa \mid aBa \mid aAa \mid aa \\ & A & \to & Bc \mid bB \mid a \mid c \mid b \\ & B & \to & Bc \mid bB \mid c \mid b \end{array}$$

(a) Převod vlastní gramatiky G_8 na ekvivalentní gramatiku G_G v Greibachové normální formě.

Nejprve musíme vlastní gramatiku G_8 převést na ekvivalentní vlastní gramatiku bez levé rekurze G_9 s použitím algoritmu **4.6** ze studijní opory:

Krok (1): Položme $A_1 = S$, $A_2 = A$, $A_3 = B$, $i = 1$, $n = 3$
Krok (2): beze změny
Krok (3): $i \neq n$, takže položíme $i = i + 1 = 2$, $j = 1$
Krok (4): beze změny
Krok (5): $j = i - 1$, takže přejdeme na krok (2)
Krok (2): beze změny
Krok (3): $i \neq n$, takže položíme $i = i + 1 = 3$, $j = 1$
Krok (4): beze změny
Krok (5): $j \neq i - 1$, takže položíme $j = j + 1 = 2$ a přejdeme na krok (4)
Krok (4): beze změny
Krok (5): $j = i - 1$, takže přejdeme na krok (2)
Krok (2): Nahradíme pravidla:

$$\begin{array}{rcl} B & \rightarrow & Bc \mid bB \mid c \mid b \\ & \downarrow & \\ B & \rightarrow & bB \mid c \mid b \mid bBB' \mid cB' \mid bB' \\ B' & \rightarrow & c \mid cB' \end{array}$$

Krok (3):

$i = n$, získali jsme výslednou vlastní gramatiku bez levé rekurze:

$G_9 = (\{S, A, B, B'\}, \{a, b, c\}, P_9, S)$, kde:

$$\begin{array}{rrcl} P_9: & S & \rightarrow & aABa \mid aBa \mid aAa \mid aa \\ & A & \rightarrow & Bc \mid bB \mid a \mid c \mid b \\ & B & \rightarrow & bB \mid c \mid b \mid bBB' \mid cB' \mid bB' \\ & B' & \rightarrow & c \mid cB' \end{array}$$

Nyní převedeme vlastní gramatiku bez levé rekurze G_9 na ekvivalentní gramatiku G_G v Greibachové normální formě s pomocí algoritmu **4.8** ze studijní opory:

Krok (1): $S < A < B < B'$, $n = 4$
Krok (2): Položíme $i = n - 1 = 3$
Krok (3): $i \neq 0$, není co nahradit
Krok (4): Položíme $i = i - 1 = 2$, pokračujeme krokem (3)
Krok (3): $i \neq 0$, nahradíme pravidlo:

$$\begin{array}{rcl} A & \rightarrow & Bc \\ & \downarrow & \\ A & \rightarrow & bBc \mid cc \mid bc \mid bBB'c \mid cB'c \mid bB'c \end{array}$$

Krok (4): Položíme $i = i - 1 = 1$, pokračujeme krokem (3)
Krok (3): $i \neq 0$, není co nahradit
Krok (4): Položíme $i = i - 1 = 0$, pokračujeme krokem (3)
Krok (3): $i = 0$, pokračujeme krokem (5)
Krok (5): nahradíme pravidla:

$$\begin{array}{rcl} S & \rightarrow & aABa \mid aBa \mid aAa \mid aa \\ A & \rightarrow & bBc \mid cc \mid bc \mid bBB'c \mid cB'c \mid bB'c \\ & \downarrow & \\ S & \rightarrow & aABA' \mid aBA' \mid aAA' \mid aA' \\ A & \rightarrow & bBC' \mid cC' \mid bC' \mid bBB'C' \mid cB'C' \mid bB'C' \end{array}$$

Krok (6): přidáme pravidla:

$$\begin{array}{rcl} A' & \rightarrow & a \\ C' & \rightarrow & c \end{array}$$

Výsledná gramatika v Greibachové normální formě:

$G_G = (\{S, A, A', B, B', C'\}, \{a, b, c\}, P_G, S)$, kde:

$$P_G : \quad \begin{array}{rcl} S & \rightarrow & aABA' \mid aBA' \mid aAA' \mid aA' \\ A & \rightarrow & bBC' \mid cC' \mid bC' \mid bBB'C' \mid cB'C' \mid bB'C' \mid bB \mid a \mid c \mid b \\ A' & \rightarrow & a \\ B & \rightarrow & bB \mid c \mid b \mid bBB' \mid cB' \mid bB' \\ B' & \rightarrow & c \mid cB' \\ C' & \rightarrow & c \end{array}$$

(b) Převod vlastní gramatiky bez jednoduchých pravidel G_8 na ekvivalentní gramatiku G_{CH} v Chomského normální formě s pomocí algoritmu **4.7** ze studijní opory:

Krok (1): P_{CH} obsahuje pravidla:

$$\begin{aligned} A &\to a \mid c \mid b \\ B &\to c \mid b \end{aligned}$$

Krok (2): beze změny
Krok (3): beze změny
Krok (4):

$$\begin{aligned} S &\to aABa \\ &\downarrow \\ S &\to a'\langle ABa\rangle \\ \langle ABa\rangle &\to A\langle Ba\rangle \\ \langle Ba\rangle &\to Ba' \end{aligned} \qquad \begin{aligned} S &\to aBa \\ &\downarrow \\ S &\to a'\langle Ba\rangle \\ \langle Ba\rangle &\to Ba' \end{aligned} \qquad \begin{aligned} S &\to aAa \\ &\downarrow \\ S &\to a'\langle Aa\rangle \\ \langle Aa\rangle &\to Aa' \end{aligned}$$

Krok (5):

$$\begin{aligned} S &\to aa \\ &\downarrow \\ S &\to a'a' \end{aligned} \qquad \begin{aligned} A &\to Bc \\ &\downarrow \\ A &\to Bc' \end{aligned} \qquad \begin{aligned} A &\to bB \\ &\downarrow \\ A &\to b'B \end{aligned} \qquad \begin{aligned} B &\to Bc \\ &\downarrow \\ B &\to Bc' \end{aligned} \qquad \begin{aligned} B &\to bB \\ &\downarrow \\ B &\to b'B \end{aligned}$$

Krok (6): Do P_{CH} přidáme pravidla:

$$\begin{aligned} a' &\to a \\ b' &\to b \\ c' &\to c \end{aligned}$$

Výsledná gramatika v Chomského normální formě:

$G_{CH} = (\{S, A, B, \langle ABa\rangle, \langle Ba\rangle, \langle Aa\rangle, a', b', c'\}, \{a, b, c\}, P_{CH}, S)$, kde:

$$\begin{aligned} P_{CH}: \qquad S &\to a'\langle ABa\rangle \mid a'\langle Ba\rangle \mid a'\langle Aa\rangle \mid a'a' \\ A &\to a \mid c \mid b \mid Bc' \mid b'B \\ B &\to c \mid b \mid Bc' \mid b'B \\ \langle ABa\rangle &\to A\langle Ba\rangle \\ \langle Ba\rangle &\to Ba' \\ \langle Aa\rangle &\to Aa' \\ a' &\to a \\ b' &\to b \\ c' &\to c \end{aligned}$$