# Úkol č. 2 do předmětu Teoretická informatika

Vojtěch Havlena (xhavle03)

## 1. Příklad

– Předpokládejme, že jazyk $L$ je bezkontextový. Potom podle Pumping lemma pro bezkontextové jazyky existuje konstanta $k > 0$ taková, že $\forall z \in L \wedge |z| \geq k \Rightarrow z = uvwxy \wedge vx \neq \varepsilon \wedge |vwx| \leq k \wedge uv^iwx^iy \in L$ pro každé $i \geq 0$.

– Zvolme libovolné takové $k > 0$, splňující výše uvedené. Dále zvolme řetězec $z = a^kb^kc^kd^k \in L$, přičemž $|z| = 4k \geq k$.

– Z řetězců $u, v, w, x, y \in \Sigma^*$, které splňují Pumping lemma zvolme libovolné z nich ($z = uvwxy$, $|vwx| \leq k$ a $vx \neq \varepsilon$). Libovolný výběr těchto řetězců musí spadnout do nějaké z následujících kategorií.

(a) $vwx \in \{a\}^+$: Vzhledem k tomu, že $vx \neq \varepsilon$, tak pro $i \geq 2$ řetězec $z' = uv^iwx^iy \notin L$, protože počet symbolů $a$ v řetězci $z'$ je alespoň o 1 větší, než je počet symbolů $c$ v řetězci $z'$.

(b) $vwx \in \{a\}^+.\{b\}^+$: V případě, kdy $v$ či $x$ obsahuje symboly $a$ i $b$, řetězec $z' = uv^iwx^iy \notin L$ pro $i \geq 2$, protože v řetězci $z'$ bude více alternací mezi podřetězci. V případě, kdy $v$ i $x$ jsou tvořeny buď pouze symboly $a$ nebo pouze symboly $b$, tak pro $i \geq 2$ řetězec $z' = uv^iwx^iy \notin L$, protože buď je počet symbolů $a$ v $z'$ větší jak počet symbolů $c$ v $z'$, nebo je počet symbolů $b$ v $z'$ větší jak počet symbolů $d$ v $z'$ (popřípadě oba zároveň). (Zvýšil se počet symbolů $a$ nebo $b$, ale počet symbolů $c$ a $d$ v $z'$ zůstává stejný).

(c) $vwx \in \{b\}^+$: Opět pro $i \geq 2$ řetězec $z' = uv^iwx^iy \notin L$, protože počet symbolů $b$ v řetězci $z'$ je alespoň o 1 větší, než je počet symbolů $d$ v řetězci $z'$.

(d) $vwx \in \{b\}^+.\{c\}^+$: Opět v případě, kdy $v$ či $x$ obsahuje symboly $b$ i $c$, řetězec $z' = uv^iwx^iy \notin L$ pro $i \geq 2$, protože v řetězci $z'$ bude více alternací mezi podřetězci. V případě, kdy $v$ i $x$ jsou tvořeny buď pouze symboly $b$ nebo pouze symboly $c$, tak pro $i \geq 2$ řetězec $z' = uv^iwx^iy \notin L$, protože buď je počet symbolů $b$ v $z'$ větší jak počet symbolů $d$ v $z'$, nebo je počet symbolů $c$ v $z'$ větší jak počet symbolů $a$ v $z'$ (popřípadě oba zároveň). (Zvýšil se počet symbolů $b$ nebo $c$, ale počet symbolů $a$ a $d$ v $z'$ zůstává stejný).

(e) $vwx \in \{c\}^+$: Pro $i \geq 2$ řetězec $z' = uv^iwx^iy \notin L$, protože počet symbolů $c$ v řetězci $z'$ je alespoň o 1 větší, než je počet symbolů $a$ v řetězci $z'$.

(f) $vwx \in \{c\}^+.\{d\}^+$: Opět v případě, kdy $v$ či $x$ obsahuje symboly $c$ i $d$, řetězec $z' = uv^iwx^iy \notin L$ pro $i \geq 2$, protože v řetězci $z'$ bude více alternací mezi podřetězci. V případě, kdy $v$ i $x$ jsou tvořeny buď pouze symboly $c$ nebo pouze symboly $d$, tak pro $i \geq 2$ řetězec $z' = uv^iwx^iy \notin L$, protože buď je počet symbolů $c$ v $z'$ větší jak počet symbolů $a$ v $z'$, nebo je počet symbolů $d$ v $z'$ větší jak počet symbolů $b$ v $z'$ (popřípadě oba zároveň). (Zvýšil se počet symbolů $c$ nebo $d$, ale počet symbolů $a$ a $b$ v $z'$ zůstává stejný).

(g) $vwx \in \{d\}^+$: Pro $i \geq 2$ řetězec $z' = uv^iwx^iy \notin L$, protože počet symbolů $d$ v řetězci $z'$ je alespoň o 1 větší, než je počet symbolů $b$ v řetězci $z'$.

– Pro žádný výběr řetězců $u, v, w, x, y$ nejsme schopni dosáhnout toho, aby $z = uvwxy \wedge vx \neq \varepsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L$, což je spor. Jazyk $L$ tedy není bezkontextový.

## 2. Příklad

V celém příkladu pracuji s následujícím vyjádřením rozdílu množin: $L \setminus L' = \overline{L'} \cap L$. Dále pokud nebude řečeno jinak, všechny zmíněné věty pocházení z opory k předmětu TIN.

(a) $L \in \mathcal{L}_3, L' \in \mathcal{L}_3$: Podle věty 3.23 jsou regulární jazyky uzavřeny vůči průniku i doplňku, tedy i $\overline{L'} \cap L = L \setminus L'$ bude regulární jazyk a tím pádem i bezkontextový jazyk.

(b) $L \in \mathcal{L}_3, L' \in \mathcal{L}_2$: Podle věty 4.24 bezkontextové jazyky nejsou uzavřeny vůči doplňku a tedy $\overline{L'}$ není nutně bezkontextový jazyk. Nyní předpokládejme, že $L, L'$ jsou jazyky nad abecedou $\Sigma$, potom stačí zvolit $L = \Sigma^* \in \mathcal{L}_3$, tedy $L \setminus L' = \overline{L'} \cap L = \overline{L'}$, což podle předchozí věty není nutně bezkontextový jazyk. A tedy $L \setminus L'$ není nutně bezkontextový jazyk.

(c) $L \in \mathcal{L}_2, L' \in \mathcal{L}_3$: Podle věty 3.23 jsou regulární jazyky uzavřeny vůči doplňku, tedy $\overline{L'} \in \mathcal{L}_3$. Dále podle věty 4.22 jsou bezkontextové jazyky uzavřeny vzhledem k průniku s regulárními jazyky, tedy $\overline{L'} \cap L = L \setminus L'$ je nutně bezkontextový jazyk.

(d) $L \in \mathcal{L}_2, L' \in \mathcal{L}_2$: Podle věty 4.24 bezkontextové jazyky nejsou uzavřeny vůči doplňku a průniku. Jazyk $\overline{L'}$ tedy není nutně bezkontextový. Nyní opět předpokládejme, že $L, L'$ jsou jazyky nad abecedou $\Sigma$, potom stačí zvolit $L = \Sigma^* \in \mathcal{L}_2$, tedy $L \setminus L' = \overline{L'} \cap L = \overline{L'}$, což podle předchozí věty není nutně bezkontextový jazyk. A tedy $L \setminus L'$ není nutně bezkontextový jazyk.

## 3. Příklad

(a) Algoritmus pro výpočet minimální váhy je založen na iterativním výpočtu minimální váhy řetězce, který lze vyderivovat z každého neterminálního symbolu v gramatice $G$, převedenou do Chomského normální formy. Tato minimální váha pro každý neterminál je označena jako $M_A$, $A \in N$.

Hlavní myšlenka algoritmu je následující: Na začátku je pro každý neterminál nastavena hodnota $M_A := \infty$. Pro pravidla gramatiky ve tvaru $A \to x$,

$x \in (\Sigma \cup \{\varepsilon\})$, je možné $M_A$ spočítat přímo jako $M_A = \min(M_A, ||x||)$. Jakmile již došlo k nastavení hodnoty $M_A < \infty$, tak je možné uvažovat i pravidla, která obsahují na pravé straně právě tyto neterminály, které již mají předběžně spočítanou minimální váhu. Potom tedy pro pravidla tvaru $A \to BC$, $A, B, C \in N$, se $M_A$ spočítá jako $M_A = \min(M_A, M_B + M_C)$. V případě, že je hodnota $M_A$ aktualizována (tj. nastavena na menší váhu), tak je nutné znovu přepočítat váhu neterminálů, které vystupují na levé straně pravidlech obsahující $A$ na pravé straně pravidla. Tedy pokud aktualizuji hodnotu $M_A$ je nutné přepočítat váhu všech neterminálů $B$, takových, že $B \to AX$ nebo $B \to XA$, $X, B, A \in N$, jsou pravidla z $P$ (toto zajišťuje procedura Aktualizuj). Tento postup se opakuje, dokud minimální váha není spočítána pro každý neterminál. Potom výstupem algoritmu je hodnota $M_S$, kde $S$ je výchozí symbol gramatiky.

**Procedura** Aktualizuj($P$, $B$)

**foreach** $A \to X_1X_2 \in P$, *kde* $X_1 = B \vee X_2 = B$, $X_1, X_2, A \in N'$ **do**
    **if** $M_A > M_{X_1} + M_{X_2}$ **then**
        $M_A := M_{X_1} + M_{X_2}$
        Aktualizuj($P$, $A$)
    **end**
**end**

**Algoritmus 1:** Minimální váha gramatiky, $\Sigma \subset \mathbb{N}$

**Vstup**: Bezkontextová gramatika $G = (N, \Sigma, P, S)$
**Výstup**: Váha bezkontextové gramatiky, $||G||$

Převedení $G$ na ekvivalentní gramatiku $\overline{G}$ bez zbytečných symbolů.
Převedení gramatiky $\overline{G}$ na ekvivalentní gramatiku $G'$ v Chomského normální formě, $G' = (N', \Sigma', P', S)$
Polož $M_A := \infty$ pro každé $A \in N'$
Polož $N_0 := \emptyset$, $P_0 = \emptyset$, $i := 1$
**repeat**
    $N_i := \{A | A \to \alpha \in P' \wedge \alpha \in (N_{i-1} \cup \Sigma)^*\} \cup N_{i-1}$
    $P_i := \{A \to \alpha | A \to \alpha \in P' \wedge \alpha \in (N_{i-1} \cup \Sigma)^*\}$
    **foreach** $A \to \alpha \in P_i$ **do**
        **if** $\alpha = x$, *kde* $x \in \Sigma'$ **and** $M_A > ||x||$ **then**
            $M_A := ||x||$
            Aktualizuj($P_i$, $A$)
        **else if** $\alpha = CD$, *kde* $C, D \in N'$ **and** $M_A > M_C + M_D$ **then**
            $M_A := M_C + M_D$
            Aktualizuj($P_i$, $A$)
        **end**
    **end**
    $i := i + 1$
**until** $N_{i-1} = N_{i-2}$;
**return** $M_S$

(b) Pokud $\Sigma \subset \mathbb{Z}$, může nastat situace, kdy $M_A = -\infty$. Toto může nastat tehdy, pokud v $G$ existuje derivace $A \Rightarrow^* \alpha A \beta$ a $M_\alpha + M_\beta < 0$, kde $\alpha, \beta \in (N \cup \Sigma)^*$ a $M_\alpha$ je součet minimálních vah symbolů v řetězci $\alpha$. Pro detekci těchto záporných derivací upravíme proceduru Aktualizuj. Tuto proceduru rozšíříme o možnost uložení (množina $T$) všech pravidel, které vedly ke snížení hodnoty $M_A$. Pokud použijeme pravidlo, které se již v množině $T$ vyskytuje a pokud navíc dojde opět ke snížení hodnoty $M_A$, znamená to, že $A \Rightarrow^* \alpha A \beta$ a navíc došlo ke snížení hodnoty $M_A$, tedy $M_A > M_A + M_\alpha + M_\beta$. Což znamená, že $M_\alpha + M_\beta < 0$ a vznikla tedy záporná rekurze a tom případě nastavíme $M_A := -\infty$.

**Procedura** Aktualizuj($P$, $B$, $T$)

**foreach** $A \to X_1 X_2 \in P$, *kde* $X_1 = B \lor X_2 = B$, $X_1, X_2, A \in N'$ **do**
  **if** $M_A > M_{X_1} + M_{X_2}$ **then**
    **if** $A \to X_1 X_2 \in T$ **then**
      $M_A := -\infty$
    **else**
      $M_A := M_{X_1} + M_{X_2}$
    **end**
    Aktualizuj($P$, $A$, $T \cup \{A \to X_1 X_2\}$)
**end**

**Algoritmus 2:** Minimální váha gramatiky, $\Sigma \subset \mathbb{Z}$

**Vstup**: Bezkontextová gramatika $G = (N, \Sigma, P, S)$

**Výstup**: Váha bezkontextové gramatiky, $||G||$

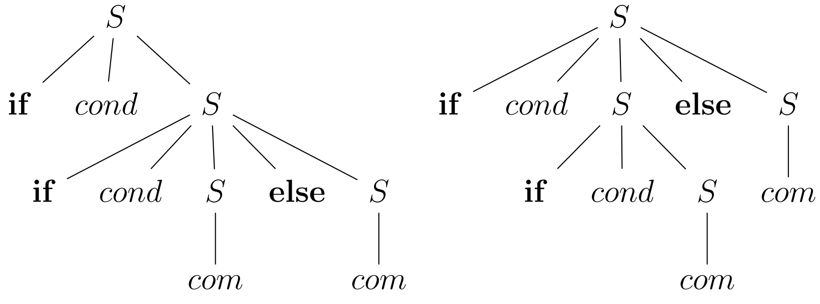
Převedení $G$ na ekvivalentní gramatiku $\overline{G}$ bez zbytečných symbolů.
Převedení gramatiky $\overline{G}$ na ekvivalentní gramatiku $G'$ v Chomského normální formě, $G' = (N', \Sigma', P', S)$
Polož $M_A := \infty$ pro každé $A \in N'$
Polož $N_0 := \emptyset$, $P_0 = \emptyset$, $i := 1$
**repeat**
  $N_i := \{A | A \to \alpha \in P' \land \alpha \in (N_{i-1} \cup \Sigma)^*\} \cup N_{i-1}$
  $P_i := \{A \to \alpha | A \to \alpha \in P' \land \alpha \in (N_{i-1} \cup \Sigma)^*\}$
  **foreach** $A \to \alpha \in P_i$ **do**
    **if** $\alpha = x$, *kde* $x \in \Sigma'$ **and** $M_A > ||x||$ **then**
      $M_A := ||x||$
      Aktualizuj($P_i$, $A$, $\{A \to \alpha\}$)
    **else if** $\alpha = CD$, *kde* $C, D \in N'$ **and** $M_A > M_C + M_D$ **then**
      $M_A := M_C + M_D$
      Aktualizuj($P_i$, $A$, $\{A \to \alpha\}$)
    **end**
  **end**
  $i := i + 1$
**until** $N_{i-1} = N_{i-2}$;
**return** $M_S$

**4. Příklad**

- Tvrzení dokážeme indukcí k počtu symbolů 0 ve slově $w \in L$ (počet symbolů 0 je označen jako $n$) .
- *Bázový případ* $n = 0$ (počet symbolů 0 je roven nule). V tomto případě $w = 0^0 1^k \in L$, kde $0 \leq k \leq 0$. Tedy $k = 0$ a $w = 0^0 1^0 = \varepsilon \in L$. V gramatice $G$ ale existuje derivace $S \Rightarrow \varepsilon$ a tedy $w = \varepsilon \in L(G)$.
- *Indukční krok.* Předpokládejme, že tvrzení platí pro všechny řetězce $w' \in L$, kde $\#_0(w') \leq n$ pro nějaké $n \geq 0$. Ukážeme, že tvrzení platí i pro řetězce $w$ takové, že $\#_0(w) = n + 1$.
  - Nejprve zvolme řetězec $w \in L$ takový, že platí $\#_0(w) = n + 1 > 0$. Tedy $w = 0^{n+1}1^k$, kde $0 \leq 2(n + 1) \leq k \leq 3(n + 1)$.
  - Podle indukčního předpokladu, pro řetězec $w' = 0^n 1^l \in L$, kde $0 \leq 2n \leq l \leq 3n$ a $\#_0(w') = n$, platí, že $w' \in L(G)$. Tím pádem existuje derivace $S \Rightarrow^* w'$.
  - Vzhledem k tomu, že $0 \leq 2n+2 \leq k$ a $0 \leq 2n \leq l$, pro $n > 0$, tak $2 \leq k-l$. Podobně z druhé části nerovnice $k \leq 3n+3$ a $l \leq 3n$ dostáváme $k-l \leq 3$. Dohromady tedy máme $2 \leq k - l \leq 3$. V řetězci $w$ je tedy o 2 nebo o 3 více symbolů 1 než v řetězci $w' \in L(G)$. Řetězec $w$ tedy můžeme zapsat následovně: $w = 0w'11 = 00^n 1^l 11$ nebo $w = 0w'111 = 00^n 1^l 111$. Nyní budeme postupně zkoumat oba případy.
    (a) Nejprve případ, kdy $w = 00^n 1^l 11$. V gramatice $G$ máme pravidlo $S \to 0S11$. S využitím výše uvedeného dostáváme $S \Rightarrow 0S11 \Rightarrow^* 00^n 1^l 11 = w$ a tedy $S \Rightarrow^* w$ a tudíž $w \in L(G)$.
    (b) Nyní případ, kdy $w = 00^n 1^l 111$. V gramatice $G$ máme pravidlo $S \to 0S111$. Opět s využitím výše uvedeného dostáváme $S \Rightarrow 0S111 \Rightarrow^* 00^n 1^l 111 = w$ a tedy $S \Rightarrow^* w$ a tudíž opět $w \in L(G)$.
- Dokázali jsme tedy, že $L \subseteq L(G)$.

**5. Příklad**

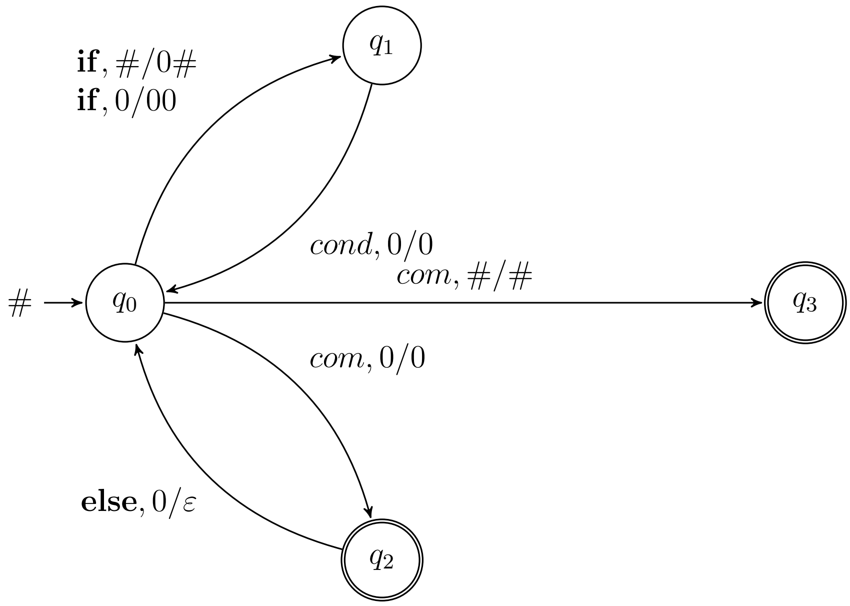
(a) Gramatika není jednoznačná, je víceznačná, protože generuje víceznačnou větu $w =$ **if** *cond* **if** *cond com* **else** *com*. Této větě odpovídají dva derivační stromy:



(b) Jazyk $L(G)$ není jazyk s inherentní víceznačností, protože pro něj existuje jednoznačná bezkontextová gramatika $G'$. Tato gramatika je definována jako $G' = (\{S, B\}, \{\mathbf{if}, \mathbf{else}, \mathit{cond}, \mathit{com}\}, P, S)$ s pravidly

$$\begin{aligned} S &\rightarrow \mathbf{if}\ \mathit{cond}\ S \mid \mathbf{if}\ \mathit{cond}\ B\ \mathbf{else}\ S \mid \mathit{com} \\ B &\rightarrow \mathbf{if}\ \mathit{cond}\ B\ \mathbf{else}\ B \mid \mathit{com} \end{aligned}$$

(c) Výsledný deterministický zásobníkový automat přijímající jazyk $L(G)$:




Přijetí slova $w = \mathbf{if}\ \mathit{cond}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}\ \mathbf{else}\ \mathit{com}\ \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}$:

$$\begin{aligned}
(q_0, w, \#) &\vdash (q_1, \mathit{cond}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}\ \mathbf{else}\ \mathit{com}\ \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}, 0\#) \vdash \\
&\vdash (q_0, \mathbf{if}\ \mathit{cond}\ \mathit{com}\ \mathbf{else}\ \mathit{com}\ \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}, 0\#) \vdash \\
&\vdash (q_1, \mathit{cond}\ \mathit{com}\ \mathbf{else}\ \mathit{com}\ \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}, 00\#) \vdash \\
&\vdash (q_0, \mathit{com}\ \mathbf{else}\ \mathit{com}\ \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}, 00\#) \vdash \\
&\vdash (q_2, \mathbf{else}\ \mathit{com}\ \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}, 00\#) \vdash \\
&\vdash (q_0, \mathit{com}\ \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}, 0\#) \vdash (q_2, \mathbf{else}\ \mathbf{if}\ \mathit{cond}\ \mathit{com}, 0\#) \vdash \\
&\vdash (q_0, \mathbf{if}\ \mathit{cond}\ \mathit{com}, \#) \vdash (q_1, \mathit{cond}\ \mathit{com}, 0\#) \vdash (q_0, \mathit{com}, 0\#) \vdash \\
&\vdash (q_2, \varepsilon, 0\#)
\end{aligned}$$