Teoreticka informatika (TIN) – 2022/2023
Úkol 2

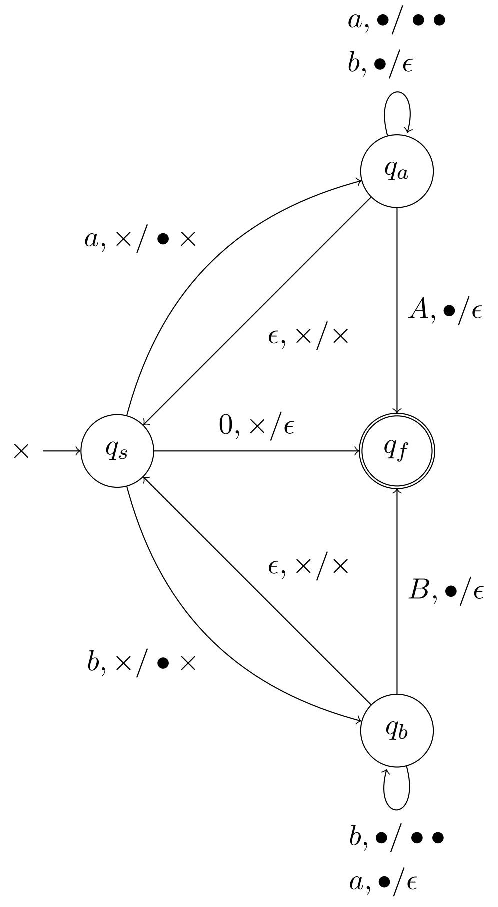
Domink Nejedlý (xnejed09)

1. Uvažujte abecedu $\Sigma = \{a, b, A, 0, B\}$. Sestrojte deterministický zásobníkový automat přijímající jazyk $L \subseteq \Sigma^*$ definovaný jako

$$\begin{aligned} L = &\{w.0 \mid w \in \{a,b\}^* \wedge \#_a(w) = \#_b(w)\} \cup \\ &\{w.A \mid w \in \{a,b\}^* \wedge \#_a(w) > \#_b(w)\} \cup \\ &\{w.B \mid w \in \{a,b\}^* \wedge \#_a(w) < \#_b(w)\}, \end{aligned}$$

kde $\#_x(w)$ pro $x \in \{a, b\}$ značí počet výskytů symbolu x v řetězci w. Použijte abecedu zásobníkových symbolů $\Gamma = \{\times, \bullet\}$ a symbol $\times$ jako startovací symbol zásobníku. Automat zapište v grafické formě. Demonstrujte přijetí slova $abaabA$.

Řešení: Deterministický zásobníkový automat (DZA) M přijímající jazyk L ($L(M) = L$):



Obrázek 1: DZA M

Slovo $abaabA$ přijme M následující sekvencí přechodů:

$$\begin{aligned} (q_s, abaabA, \times) &\vdash_M (q_a, baabA, \bullet\times) \vdash_M (q_a, aabA, \times) \vdash_M (q_s, aabA, \times) \vdash_M (q_a, abA, \bullet\times) \\ &\vdash_M (q_a, bA, \bullet\bullet\times) \vdash_M (q_a, A, \bullet\times) \vdash_M (q_f, \epsilon, \times) \end{aligned}$$

2. Uvažujte abecedu Σ takovou, že $\Sigma \cap \{0,1\} = \emptyset$ a jazyk $L = L(G)$ nad abecedou Σ zadaný pomocí bezkontextové gramatiky G. Dále uvažujte jazyk L' definovaný jako

$$L' = \{w.x \mid w \in \Sigma^* \wedge (w \in L \iff x = 1) \wedge (w \notin L \iff x = 0)\}.$$

Inspirujte se v kapitole o uzávěrových vlastnostech bezkontextových jazyků a proveďte jedno z následujících:

(a) Navrhněte algoritmus, který bude mít na vstupu gramatiku G a sestaví bezkontextovou gramatiku G' takovou, že $L(G') = L'$.

(b) Dokažte, že takový algoritmus neexistuje.

Řešení: Takový algoritmus neexistuje. Zřejmě

$$L' = \{w.x \mid w \in \Sigma^* \wedge (w \in L \iff x = 1) \wedge (w \notin L \iff x = 0)\} = L\{1\} \cup \overline{L}\{0\}.$$

Uvažme abecedu $\Sigma = \{a, b, c\}$ a jazyk $L = \Sigma^* \setminus \{a^n b^n c^n \mid n \geq 1\} = \overline{\{a^n b^n c^n \mid n \geq 1\}} \in \mathcal{L}_2$. Pak

$$L' = \overline{\{a^n b^n c^n \mid n \geq 1\}}\{1\} \cup \{a^n b^n c^n \mid n \geq 1\}\{0\}.$$

Nyní ukážeme, že $L' \notin \mathcal{L}_2$. Důkaz provedeme sporem pomocí Pumping lemmatu pro bezkontextové jazyky. Předpokládejme, že $L' \in \mathcal{L}_2$. Potom dle Pumping lemmatu $\exists k > 0 : \forall z \in L' : |z| \geq k \implies \exists u, v, w, x, y \in \Sigma^* : z = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L'$. Uvažme libovolné k splňující výše uvedenou podmínku. Zvolme slovo $z = a^k b^k c^k 0 \in L'$. Zřejmě $|z| = 3k + 1 \geq k$. Uvažme libovolné $u, v, w, x, y \in \Sigma^*$ splňující $z = a^k b^k c^k 0 = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k$. V souladu s těmito podmínkami jsou možné následující dekompozice:

- Řetězec $vwx = a^m b^n$, kde $m, n \geq 0$ a $1 \leq m + n \leq k$. Pak ale pro všechna $i \neq 1$ platí, že $uv^iwx^iy \notin L'$, jelikož $\#_a(uv^iwx^iy) \neq \#_c(uv^iwx^iy)$ nebo $\#_b(uv^iwx^iy) \neq \#_c(uv^iwx^iy)$ a současně slovo uv^iwx^iy nekončí symbolem 1 (1 není sufixem uv^iwx^iy).
- Řetězec $vwx = b^m c^n$, kde $m, n \geq 0$ a $1 \leq m + n \leq k$. Pak ale pro všechna $i \neq 1$ platí, že $uv^iwx^iy \notin L'$, jelikož $\#_a(uv^iwx^iy) \neq \#_b(uv^iwx^iy)$ nebo $\#_a(uv^iwx^iy) \neq \#_c(uv^iwx^iy)$ a současně slovo uv^iwx^iy nekončí symbolem 1 (1 není sufixem uv^iwx^iy).
- Řetězec $vwx = c^n 0$, kde $0 \leq n \leq k - 1$. Potom ale při volbě $i = 0$ slovo uv^iwx^iy nekončí ani jedním ze symbolů 0 a 1 (0 ani 1 není sufixem uv^iwx^iy) nebo $\#_c(uv^iwx^iy) < \#_b(uv^iwx^iy)$, přičemž slovo uv^iwx^iy není ukončeno symbolem 1 (1 není sufixem uv^iwx^iy). Proto platí, že $uv^iwx^iy \notin L'$.

Jiné dekompozice nejsou možné, jelikož by porušovaly podmínku $|vwx| \leq k$. Vidíme tedy, že pro každý možný rozklad lze výsledné slovo uv^iwx^iy z jazyka L' "vypumpovat", což vede ke sporu s počátečním předpokladem. Z toho důvodu opravdu $L' \notin \mathcal{L}_2$. Pro jazyk L' tedy neexistuje bezkontextová gramatika G' taková, že $L(G') = L'$, a proto nemůže existovat univerzální algoritmus, který by byl schopen takové gramatiky pro libovolné bezkontextové jazyky L tvořit.

3. Navrhněte algoritmus, který pro bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ spočítá množinu

$$N_{aa} = \{A \in N \mid \exists w \in \Sigma^* : A \Rightarrow_G^+ w \wedge aa \text{ je podřetězec } w\}.$$

V algoritmu můžete využít množinu $N_\epsilon = \{A \in N \mid A \Rightarrow_G^+ \epsilon\}$. Doporučujeme nadefinovat si další vhodné množiny neterminálů a algoritmicky popsat jejich výpočet (u N_ϵ popis výpočtu není potřeba).

Ilustrujte použití algoritmu na příkladě gramatiky s pravidly

$$S \rightarrow XY \mid YX \mid Z \qquad X \rightarrow Xba \mid \epsilon \qquad Y \rightarrow Yab \mid \epsilon \qquad Z \rightarrow XbaaU$$

Řešení: Nejprve si nadefinujme pomocné množiny neterminálů a postupy jejich výpočtů. Začněme s množinou neterminálů generující terminální řetězce, označovanou jako N_t.

Algoritmus 1: Výpočet množiny N_t

Vstup: bezkontextová gramatika $G = (N, \Sigma, P, S)$
Výstup: $N_t = \{A \in N \mid \exists w \in \Sigma^* : A \Rightarrow_G^+ w\}$
1 $N_t^0 := \emptyset, i := 0$
2 $N_t^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P : \alpha \in (N_t^i \cup \Sigma)^*\}$
3 Jestliže $N_t^{i+1} = N_t^i$, pak $N_t := N_t^i$ a ukonči výpočet, jinak $i := i + 1$ a vrať se na bod 2.

Pokračujme množinou neterminálů, jež mohou generovat řetězce začínající symbolem a (a je tedy jejich prefixem), označovanou jako $N_{a\text{-}pre}$.

Algoritmus 2: Výpočet množiny $N_{a\text{-}pre}$

Vstup: bezkontextová gramatika $G = (N, \Sigma, P, S)$
Výstup: $N_{a\text{-}pre} = \{A \in N \mid \exists w \in \Sigma^* : A \Rightarrow_G^+ w \wedge a \text{ je prefix } w\}$
1 Vypočti N_ϵ.
2 Vypočti N_t pomocí algoritmu 1.
3 $N_{a\text{-}pre}^0 := \emptyset, i := 0$
4 $N_{a\text{-}pre}^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P : \alpha \in N_\epsilon^*(\{a\} \cup N_{a\text{-}pre}^i)(\Sigma \cup N_t)^*\}$
5 Jestliže $N_{a\text{-}pre}^{i+1} = N_{a\text{-}pre}^i$, pak $N_{a\text{-}pre} := N_{a\text{-}pre}^i$ a ukonči výpočet, jinak $i := i + 1$ a vrať se na bod 4.

Dále zadefinujme množinu neterminálů, jež mohou generovat řetězce končící symbolem a (a je tedy jejich sufixem), označovanou jako $N_{a\text{-}suf}$.

Algoritmus 3: Výpočet množiny $N_{a\text{-}suf}$

Vstup: bezkontextová gramatika $G = (N, \Sigma, P, S)$
Výstup: $N_{a\text{-}suf} = \{A \in N \mid \exists w \in \Sigma^* : A \Rightarrow_G^+ w \wedge a \text{ je sufix } w\}$
1 Vypočti N_ϵ.
2 Vypočti N_t pomocí algoritmu 1.
3 $N_{a\text{-}suf}^0 := \emptyset, i := 0$
4 $N_{a\text{-}suf}^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P : \alpha \in (\Sigma \cup N_t)^*(\{a\} \cup N_{a\text{-}suf}^i)N_\epsilon^*\}$
5 Jestliže $N_{a\text{-}suf}^{i+1} = N_{a\text{-}suf}^i$, pak $N_{a\text{-}suf} := N_{a\text{-}suf}^i$ a ukonči výpočet, jinak $i := i + 1$ a vrať se na bod 4.

Nyní s pomocí výše zavedených množin definujme postup výpočtu množiny N_{aa}.

Algoritmus 4: Výpočet množiny N_{aa}

Vstup: bezkontextová gramatika $G = (N, \Sigma, P, S)$
Výstup: $N_{aa} = \{A \in N \mid \exists w \in \Sigma^* : A \Rightarrow_G^+ w \wedge aa \text{ je podřetězec } w\}$
1 Vypočti N_ϵ.
2 Vypočti N_t pomocí algoritmu 1.
3 Vypočti $N_{a\text{-}pre}$ pomocí algoritmu 2.
4 Vypočti $N_{a\text{-}suf}$ pomocí algoritmu 3.
5 $N_{aa}^0 := \emptyset, i := 0$
6 $N_{aa}^{i+1} := \{A \in N \mid \exists (A \to \alpha) \in P : \alpha \in (\Sigma \cup N_t)^*((\{a\} \cup N_{a\text{-}suf})N_\epsilon^*(\{a\} \cup N_{a\text{-}pre}) \cup N_{aa}^i)(\Sigma \cup N_t)^*\}$
7 Jestliže $N_{aa}^{i+1} = N_{aa}^i$, pak $N_{aa} := N_{aa}^i$ a ukonči výpočet, jinak $i := i + 1$ a vrať se na bod 6.

Ilustrujme nyní použití algoritmu 4 na gramatice uvedené v zadání. Předpokládejme že $N = \{S, X, Y, Z, U\}$ a $\Sigma = \{a, b\}$.

(1) Výpočet N_ϵ (krok 1):

$$N_\epsilon^0 = \emptyset$$

$$N_\epsilon^1 = \{X, Y\}$$
$$N_\epsilon^2 = \{S, X, Y\} = N_\epsilon^3 = N_\epsilon$$

(2) Výpočet N_t (krok 2):

$$N_t^0 = \emptyset$$
$$N_t^1 = \{X, Y\}$$
$$N_t^2 = \{S, X, Y\} = N_t^3 = N_t$$

(3) Výpočet $N_{a\text{-}pre}$ (krok 3):

$$N_{a\text{-}pre}^0 = \emptyset$$
$$N_{a\text{-}pre}^1 = \{Y\}$$
$$N_{a\text{-}pre}^2 = \{S, Y\} = N_{a\text{-}pre}^3 = N_{a\text{-}pre}$$

(4) Výpočet $N_{a\text{-}suf}$ (krok 4):

$$N_{a\text{-}suf}^0 = \emptyset$$
$$N_{a\text{-}suf}^1 = \{X\}$$
$$N_{a\text{-}suf}^2 = \{S, X\} = N_{a\text{-}suf}^3 = N_{a\text{-}suf}$$

(5) Samotný výpočet N_{aa} (kroky 5, 6 a 7):

$$N_{aa}^0 = \emptyset$$
$$N_{aa}^1 = \{S\} = N_{aa}^2 = N_{aa}$$

4. Uvažujte následující operátor $\blacktriangle : 2^{\Sigma^*} \times 2^{\Sigma^*} \to 2^{\Sigma^*}$ na jazycích nad abecedou Σ:

$$L_1 \blacktriangle L_2 = \{w \in \Sigma^* \mid \forall w_1, w_2 \in \Sigma^* : w_1 w_2 = w \implies (w_1 \in L_1 \vee w_2 \in L_2)\}$$

Dokažte, že množina rekurzivně vyčíslitelných jazyků je uzavřena na $\blacktriangle$.

Řešení: Uvažujme dva libovolné rekurzívně vyčíslitelné (RE) jazyky L_1 a L_2 nad abecedou Σ. Jelikož $L_1, L_2 \in \mathcal{L}_0 = \mathcal{L}_{RE} = \mathcal{L}_{TS}$, existují TS (Turingův stroj) M_1 a TS M_2 takové, že $L_1 = L(M_1)$ a $L_2 = L(M_2)$. Nyní sestrojme třípáskový TS $M_{L_1 \blacktriangle L_2}$ takový, že $L(M_{L_1 \blacktriangle L_2}) = L_1 \blacktriangle L_2$. Předpokládejme, že zpracovávaný řetězec je u M na počátku zapsán na první pásce, zbylé dvě pásky jsou prázdné a všechny čtecí/zápisové hlavy jsou na nejlevější pozici. M může postupovat následujícím způsobem:

(1) M nakopíruje prefix vstupního řetězce z první pásky od jejího počátku po aktuální pozici hlavy (tedy prefix vymezený nejlevější pozicí první pásky a aktuálním umístěním čtecí/zápisové hlavy na této pásce) včetně na druhou pásku a zbytek vstupu z první pásky na pásku třetí.

(2) M prokládaně krok po kroku simuluje na druhé pásce M_1 a na třetí pásce M_2. Vždy tedy provede krok simulace M_1, poté M_2 a následně tento cyklus opakuje. V případě, že M_1, nebo M_2 zamítne svůj vstup (zasekne se či abnormálně skončí), pokračuje dále simulace pouze toho druhého (nezastaveného) z nich již bez střídání.

(3) Ve chvíli, kdy jeden ze strojů M_1 nebo M_2 přijme svůj vstup, hlava na první pásce se posune o jednu pozici doprava. Pokud se pod ní následně nachází symbol Δ (blank), tak M přijme vstupní řetězec (pro všechna jeho rozdělení na dvě části platí, že první z nich patří do L_1 nebo druhá patří do L_2). Jinak M smaže obsah druhé i třetí pásky, hlavy na těchto páskách přesune na nejlevější pozici (první symbol Δ) a přejde na bod 1.

(4) Pokud M_1 ani M_2 nepřijme, tedy pokud oba tyto stroje zamítnou nebo cyklí (případně kombinace – jeden zamítne a druhý cyklí), pak také M zamítne/cyklí (tedy nepřijme).

Pro libovolné dva jazyky L_1 a L_2 nad abecedou Σ lze tedy sestrojit TS M (zde třípáskový) přijímající jazyk $L_1 \blacktriangle L_2$. Množina rekurzivně vyčíslitelných jazyků je proto na $\blacktriangle$ uzavřena.