Fakulta informačních technologií
Vysoké učení technické v Brně

TIN
Teoretická informatika

2. domáca úloha

Adrián Tóth (xtotha01) 29. novembra 2018

Obsah

1 Príklad číslo 1

1.1 (a)

Definice 4.29 [1](str. č. 97) Označme ZAV_n pre $n \geq 0$ jazyky setávající ze všech vyvážených řetězců závorek n typů. Tyto jazyky – označované též jako Dyckovy jazyky – jsou generovány gramatikami s pravidly tvaru: $S \rightarrow [^1\ S\ ]^1 \mid [^2\ S\ ]^2 \mid ... \mid [^n\ S\ ]^n \mid SS \mid \varepsilon$

Z hore uvedenej definície pre náš príklad vyplýva, že náš dyckov jazyk L je generovaný

$$S \rightarrow \varepsilon \mid SS \mid [\ S\ ]$$

Každé slovo $w \in L$, pre ktoré platí že $w \neq \varepsilon$, vieme zapísať v tvare $[u]v$ kde $u, v \in L$ pretože

$$S \Rightarrow SS \Rightarrow [S]S = [u]v$$

Keďže $S \Rightarrow^* w$ a $S \Rightarrow^* [S]S$ tak je zrejmé, že $S \Rightarrow^* u$ a $S \Rightarrow^* v$ z čoho vyplýva, že $S \Rightarrow^* [u]v = w$ kde $\forall u, v, w \in L$.

Najkratší neprázdny prefix slova w ktorý patrí do L je [ ].

1.2 (b)

Báza

$\varepsilon \in L$ pretože $S \Rightarrow^* \varepsilon$ keďže existuje pravidlo $S \rightarrow \varepsilon$

Indukčný predpoklad

$S \Rightarrow^* w$ kde $w \in L \land w = [u]v$ čo platí pre $j < i$

Pre i + 1

$S \Rightarrow^* w$ kde $w \in L$ pre ktoré platí, že $\#_{[}(w) = i + 1$.
Z definicie Dyckovho jazyka platí, že $\#_{]}(w) = i + 1$.
Potom vieme w zapísať ako $w = [u]v$ podľa bodu (a)(kapitola 1.1) $\Longrightarrow \#_{[}(u) + \#_{[}(v) = i$.
Analogicky musí platiť $\#_{]}(u) + \#_{]}(v) = i$.

Pre i

$\#_{[}(u) + \#_{[}(v) \stackrel{?}{=} i$ analogicky pre $\#_{]}(u) + \#_{]}(v) \stackrel{?}{=} i$

1.) $\#_{[}(u) = 0 \Rightarrow \#_{[}(v) = i \quad \vee \quad \#_{[}(v) = 0 \Rightarrow \#_{[}(u) = i$

Ak je buď $\#_{[}(u)$ alebo buď $\#_{[}(v)$ rovné nule, tak ho vieme vygenerovať z pravidla S na základe indukčnej bázi.

Ak je buď $\#_{[}(u)$ alebo buď $\#_{[}(v)$ rovné i, tak ten prvok prepíšeme pomocou vzorca

$$w' = [u']v' \Rightarrow \#_{[}(u') + \#_{[}(v') = i - 1, \text{ analogicky } \#_{]}(u') + \#_{]}(v') = i - 1$$

Na základe indukčného predpokladu vieme z S vygenerovať u' a v'.

$$S \Rightarrow [S]S \Rightarrow^* [u']v' = w' \text{ kde } \#_{[}(u') + \#_{[}(v') = i, \text{ analogicky } \#_{]}(u') + \#_{]}(v') = i.$$

2.) $\#_{[}(u) \neq 0 \quad \wedge \quad \#_{[}(v) \neq 0 \quad \wedge \quad \#_{[}(u) + \#_{[}(v) \leq i$

Keď $\#_{[}(u)$ a $\#_{[}(v)$ sú nenulové, tak musí platiť že

$$\#_{[}(u) < i \wedge \#_{[}(v) < i \quad \text{t.j.} \quad \#_{[}(u) = i - M \wedge \#_{[}(v) = i - N \text{ kde } M, N \in \mathbb{N} \setminus \{0\}$$

$S \Rightarrow [S]S \Rightarrow^* [u]v = w$ kde $\#_{[}(u) + \#_{[}(v) = i + 1$, analogicky $\#_{]}(u) + \#_{]}(v) = i + 1$.

2 Príklad číslo 2

Veta 4.19 [1](str. č. 92): Nechť L je bezkontextový jazyk. Pak existuje konstanta $k > 0$ taková že je-li $z \in L$ a $|z| \geq k$, pak lze z napsat ve tvaru:

$$z = uvwxy, vx \neq \varepsilon, |vwx| \leq k$$

a pro všechna $i \geq 0$ je $uv^iwx^iy \in L$.

Nech L_{primes} je bezkontextový jazyk.

Tak existuje celočíselná konštanta $k > 0$ taká, že ak $z \in L$ a $|z| \geq k$, tak

$$z = uvwxy \wedge vx \neq \varepsilon \wedge |vwx| \leq k \wedge uv^iwx^iy \in L \text{ kde } i \geq 0$$

Zvoľme prvočíslo r väčšie ako ako k t.j. $r \geq k$ kde r je prvočíslo.

Potom platí, že

$$a^r \in L \wedge |a^r| = r \text{ kde } r \geq p \implies a^r = uvwxy \wedge vx \neq \varepsilon \wedge |vwx| \leq k \wedge uv^iwx^iy \in L \text{ pre } i \geq 0$$

Nech

$$\begin{aligned} v = a^M &\Rightarrow |v| = M \\ x = a^N &\Rightarrow |x| = N \\ w = a^O &\Rightarrow |w| = O \end{aligned}$$

Tak musí platiť že $M + N > 0$ pretože $vx \neq \varepsilon$ a $k \geq M + N + O$ pretože $|vwx| \leq k$.

Zvoľme $i = r + 1$, potom

$$uv^{r+1}wx^{r+1}y \in L$$
$$|uv^{r+1}wx^{r+1}y| = |uvwxy| + |v^r| + |x^r| = r + r \cdot M + r \cdot N = r \cdot (1 + M + N) \text{ čo nie je prvočíslo}$$

A z toho vyplýva spor pretože

$$uv^{r+1}wx^{r+1}y \notin L$$

Takže jazyk L_{primes} nie je bezkontextový jazyk.

3 Príklad číslo 3

3.1 Nerozhodnuteľnosť

Problém môžeme charakterizovať jazykom L pre ktorý platí

$$L = \{ \langle M_L \rangle \mid M_L \text{ je } TS : \exists w \in \mathit{Affine} : w \in L(M_L) \}$$

Problém členstva je charakterizovaný jazykom MP pre ktorý platí

$$MP = \{ \langle M_{MP} \rangle \# w \mid M_{MP} \text{ je } TS \text{ ktorý prijme } w \}$$

Zostavíme redukciu

$$\sigma : \{0, 1, \#\}^* \longrightarrow \{0, 1\}^* \text{ z jazyka } MP \text{ na } L$$

TS M_σ implementujúci σ priradí každému vstupu $x \in \{0, 1, \#\}^*$ reťazec $\langle M_x \rangle$, kde M_x je TS, ktorý na vstupu $y \in \{0, 1\}^*$ pracuje následovne:

1. M_x zmaže svoj vstup y.
2. Zapíše na pásku reťazec x.
3. M_x posúdi, zda $x = x_1 \# x_2$ pre x_1, ktorý je kódom TS, a x_2, ktorý je kódom jeho vstupu. Pokiaľ nie, odmietne.
4. Inak M_x simuluje činnosť TS s kódom x_1 na reťazci s kódom x_2.
   - Ak x_1 prijme x_2, tak M_x prijme.
   - Ak x_1 odmietne x_2, tak M_x odmietne.
   - Inak cyklí.

M_σ je možné implementovať úplným TS. Konečne tento TS vypíše kód M_x, ktorý sa skladá zo štyroch komponent, ktoré odpovedajú vyššie uvedeným krokom. Tri z nich sú pritom konštantné (nezávisia na x) – konkrétne (1) zmazanie pásky, (2) test na dobré sformovanie instancie MP a (3) simulácia daného TS na danom vstupe (pomocou úplného TS). TS implementujúci tieto kroky, ktoré evidentne existujú, môžeme pripraviť vopred a M_σ vypíše kód spolu s kódom na predanie riadenia. Zostáva vygenerovať kód TS, ktorý zapíše na pásku dané $x = a_1 a_2 ... a_n$. To je možné ale ľahko realizovať pomocou TS $Ra_1 Ra_2 R ... Ra_n$.

Skúmajme možné jazyky TS M_x:

- $L(M_x) = \emptyset \iff$ (x nie je správne sformovaná instancia MP) alebo ($x = x_1 \# x_2$ a TS s kódom x_1 na reťazci s kódom x_2 odmietne) alebo ($x = x_1 \# x_2$ a TS s kódom x_1 na reťazci s kódom x_2 neskončí t.j. cyklí)
- $L(M_x) = \Sigma^* \iff$ (x je správne sformovaná instancia MP, kde $x = x_1 \# x_2$ a TS s kódom x_1 na reťazci s kódom x_2 prijme)

Ak $L(M_x) = \Sigma^*$ je zrejmé, že jazyk $L(M_x)$ iste obsahuje aspoň jeden reťazec ktorý patrí do jazyka *Affine*.

Teraz už ľahko ukážeme, že σ zachováva členstvo $\langle M_x \rangle \in L \Leftrightarrow L(M_x) = \Sigma^* \Leftrightarrow x = x_1 \# x_2$ kde x_1 je kód TS, ktorý zastaví na vstupe s kódom $x_2 \Leftrightarrow x \in MP$.

3.2 Čiastočná rozhodnuteľnosť

...

4 Príklad číslo 4

...

5 Literatúra

[1] M. Češka, T. Vojnar, A. Smrčka, A. Rogalewicz: *Teoretická informatika - Studijní text.* 2018-08-23, [Online; Accessed: 2018-10-15].
URL: `http://www.fit.vutbr.cz/study/courses/TIN/public/Texty/TIN-studijni-text.pdf`