

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



TIN
Teoretická informatika

2. domácí úloha

Obsah

1	Príklad číslo 1	2
1.1	(a)	2
1.2	(b)	2
2	Príklad číslo 2	4
3	Príklad číslo 3	5
3.1	Nerozhodnuteľnosť	5
3.2	Čiastočná rozhodnuteľnosť	6
4	Príklad číslo 4	7
5	Literatúra	8

1 Príklad číslo 1

1.1 (a)

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

Z hore uvedenej definície pre náš príklad vyplýva, že náš Dyckov jazyk L je generovaný gramatikou G_D , definovanou ako $G_D = (\{S'\}, \{[,]\}, P, S')$ kde množina prepisovacích pravidiel P je daná ako

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

ktorá obsahuje iba jeden typ zátvoriek ktorými sú $[$ a $]$.

Pre každé slovo $w \in L$, pre ktoré platí že $w \neq \varepsilon$, muselo byť aspoň raz použité pravidlo $S' \rightarrow [S']$ v derivácii. S využitím pravidla $S' \rightarrow [S']$ vygenerujeme jeden pár zátvoriek, pričom medzi zátvorkami sa nachádza neterminál S' , ktorým je možné ďalej aplikovať ďalšie pravidlá a generovať reťazec u – presnejšie reťazec u patriaci do jazyka L . V prípade použitia pravidla $S' \rightarrow S'S'$ pred pravidlom $S' \rightarrow [S']$ vieme generovať ďalší reťazec na pravej strane čo odpovedá reťazcu v ktorý patrí do L , t.j. vieme generovať za $[u]$ reťazec v patriaci do jazyka L .

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

$$S' \xRightarrow{G_D} S'S' \xRightarrow{G_D} [S']S' \xRightarrow{G_D}^* [u]v$$

Keďže $S' \xRightarrow{G_D}^* u$ a $S' \xRightarrow{G_D}^* v$ ktoré patria do jazyka L , tak $S' \xRightarrow{G_D}^* [u]v$ tiež patrí do jazyka, keďže $u, v \in L$ a $[,] \in L$. Je zrejmé, že ak $S' \xRightarrow{G_D}^* w$ a $S' \xRightarrow{G_D}^* [u]v$ tak potom platí že $S' \xRightarrow{G_D}^* w = [u]v$ t.j. $w = [u]v$.

1.2 (b)

Báza

Báza je báзовý prípad **pre $i = 0$** .

Pre $i = 0$ platí, že počet $[$ a počet $]$ v reťazci w je rovno nule z čoho vyplýva, že reťazec w sa musí rovnať ε . Formálne, pre $i = 0$ platí, že $\#_[(w) = 0 \wedge \#_](w) = 0$ z čoho vyplýva, že $w = \varepsilon$.

Keďže, existuje pravidlo $S' \rightarrow \varepsilon$ v gramatike G_D a taktiež, existuje pravidlo $S \rightarrow \varepsilon$ v gramatike G , potom existujú derivácie $S' \xRightarrow{G_D} \varepsilon$ a $S \xRightarrow{G} \varepsilon$ a tak platí, že $w = \varepsilon \wedge w \in L \wedge w \in L(G)$.

Indukčný predpoklad

$S \xRightarrow{*} w$ kde $w \in L \wedge w \in L(G) \wedge w = [u]v$ čo platí pre všetky j kde $j < i$ pričom j, i značí počet zátvoriek typu $[$ a počet zátvoriek typu $]$.

Pre i

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

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

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

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

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

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

$$S \Rightarrow [S]S \Rightarrow^* [u']v' = w'$$

kde

$$\#_{\lceil}(u') + \#_{\lceil}(v') = i, \text{ analogicky } \#_{\rceil}(u') + \#_{\rceil}(v') = i$$

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

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

$$\#_{\lceil}(u) < i \wedge \#_{\lceil}(v) < i$$

t.j.

$$\#_{\lceil}(u) = i - M = j_1 \wedge \#_{\lceil}(v) = i - N = j_2 \text{ kde } M, N \in \mathbb{N} \setminus \{0\}$$

Pre i + 1

$S \Rightarrow^* w$ kde $w \in L$ pre ktoré platí, že

$$\#_{\lceil}(w) = i + 1$$

a z definície Dyckovho jazyka platí, že

$$\#_{\rceil}(w) = i + 1$$

Potom vieme w zapísať ako $w = [u]v$ podľa bodu (a) (kapitola 1.1) $\Rightarrow \#_{\lceil}(u) + \#_{\lceil}(v) = i$, analogicky musí platiť $\#_{\rceil}(u) + \#_{\rceil}(v) = i$.

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

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 konstanta $k > 0$ 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 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

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

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

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

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

1. M_x zmaže svoj vstup y .
2. Zapiše 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 instance 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 zapiše na pásku dané $x = a_1 a_2 \dots a_n$. To je možné ale ľahko realizovať pomocou $TS Ra_1 Ra_2 R \dots Ra_n$.

Skúmame možné jazyky $TS M_x$:

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

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 $\langle M_x \rangle \in L \iff L(M_x) = \Sigma^* \iff x = x_1 \# x_2$ kde x_1 je kód TS , ktorý zastaví na vstupe s kódom $x_2 \iff x \in MP$.

3.2 Čiastočná rozhodnuteľnosť

Majme $TS\ M$ pre ktorý platí, že $\exists w \in Affine : w \in L(M)$.

K čiastočnému rozhodovaniu uvedeného problému môžeme využiť $TS\ M'$ ktorý na svojej prvej páske simuluje beh $TS\ M$ pre jednotlivé možné vstupné reťazce napríklad v **lexikografickom** usporiadaní (viz dole uvedený príklad) a na druhej páske vykonáva kontrolu podmienky patričnosti reťazca do jazyka *Affine*.

$konf_M(\varepsilon, 0)$
 $konf_M(\varepsilon, 1) \# konf_M(0, 0)$
 $konf_M(\varepsilon, 2) \# konf_M(0, 1) \# konf_M(1, 0)$
 $konf_M(\varepsilon, 3) \# konf_M(0, 2) \# konf_M(1, 1) \# konf_M(00, 0)$
 $konf_M(\varepsilon, 4) \# konf_M(0, 3) \# konf_M(1, 2) \# konf_M(00, 1) \# konf_M(01, 0)$
...

M' nemôže len systematicky generovať vstupy pre $TS\ M$ a na nich nechať $TS\ M$ neobmedzene bežať pretože hrozí zacyklenie.

$TS\ M'$ ale môže mať na svojej páske súčasne rozbehnutú simuláciu $TS\ M$ pre ľubovoľný počet vstupných reťazcov, jednotlivé konfigurácie pásky budú vhodne oddelené.

$TS\ M'$ môže vždy prejsť všetky aktuálne rozbehnuté simulácie a na každej vykonať práve jeden krok. Pokiaľ v niektorom prípade dôjde k prijatiu reťazca $TS\ M$, vykoná kontrolu pratričnosti prijatého reťazca do jazyka *Affine* na druhej páske. Ak kontrola patričnosti prijatého reťazca skončila úspechom, t.j. reťazec bol prijatý $TS\ M$ a súčasne patrí do jazyka *Affine* môžeme tvrdiť, že existuje reťazec ktorý patrí do *Affine* a súčasne do $L(M)$. Ak $TS\ M$ neprijal reťazec alebo reťazec nepatrí do jazyka *Affine*, tak $TS\ M'$ pridá páskovú konfiguráciu pre ďalší reťazec a celý krok opakuje.

$TS\ M'$ vykonáva kontrolu patričnosti reťazca do *Affine* na druhej páske tak, že **TODO**.

Funkčnosť $TS\ M'$ môžeme chápať tak, že najskôr $TS\ M'$ spustí simuláciu $TS\ M$ na prvej páske. Ak $TS\ M$ zastaví a prijme určitý reťazec, tak tento reťazec $TS\ M'$ overí na druhej páske zda patrí do jazyka *Affine* t.j. skontroluje podmienku patričnosti tohto reťazca do jazyka *Affine*.

4 Příklad 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>