Úloha č.2

1. Uvažujte jazyk $L = \{w \in \{a,b\}^* \mid \#_a(w) = \#_b(w)\}$, kde $\#_x(w)$ značí počet výskytov symbolu x v reťazci w. Dokážte, že jazyk L je bezkontextový. Postupujte nasledovne:
   (a) Najprv navrhnite gramatiku G, ktorá bude mať za cieľ jazyk L generovať.
   (b) Potom pomocou indukcie k dĺžke slova $w \in L$ dokážte, že $L = L(G)$.

Riešenie

(a) $G = (N, \Sigma, P, S)$,
kde $N = \{S\}$, $\Sigma = \{a, b, \varepsilon\}$, $P = \{S \to aSb,\ S \to bSa,\ S \to SS,\ S \to \varepsilon\}$

(b) Aby sme dokázali, že $L = L(G)$, potrebujeme dokázať, že platí:

$$L(G) \subseteq L \quad \wedge \quad L \subseteq L(G)$$

(i) $\mathbf{L(G) \subseteq L}$
Pre dôkaz, že každé slovo w generované gramatikou G patrí do jazyka L použijeme indukciu vzhľadom k dĺžke n slova w:

Bázový prípad: $n = 0$, slovo $w = \varepsilon$ možno vygenerovať 2 spôsobmi:
1) priamo: $S \underset{G}{\Rightarrow} \varepsilon = w$
2) nepriamo: $S \underset{G}{\Rightarrow} SS \overset{*}{\underset{G}{\Rightarrow}} \varepsilon = w$
Keďže platí $\#_a(\varepsilon) = \#_b(\varepsilon)$, slovo $w = \varepsilon \in L$.

Indukčný predpoklad: Predpokladajme, že $L(G) \subseteq L$ platí pre všetky slová dĺžky n, takže platí: $\forall w' \Big(S \overset{*}{\underset{G}{\Rightarrow}} w' \wedge |w'| = n \implies w' \in L \Big)$

Indukčný krok: Ukážeme, že implikácia platí aj pre slová dĺžky $n+2$ (gramatika G generuje slová párnej dĺžky), tj. že platí
$\forall w \Big(S \overset{*}{\underset{G}{\Rightarrow}} w \wedge |w| = n+2 \implies w \in L \Big)$

Pri generovaní slova w pomocou gramatiky G môže nastať niekoľko prípadov, v závislosti od voľby pravidla pre prvý derivačný krok:

- $S \to aSb$: $S \underset{G}{\Rightarrow} aSb \overset{*}{\underset{G}{\Rightarrow}} aw'b = w \qquad |w'| = n,\ |w| = n+2$
  Podľa indukčného predpokladu $w' \in L$. Je zrejmé, že ak platí $\#_a(w') = \#_b(w')$, tak rovnaký vzťah platí aj pre $w = aw'b$ (pomer počtu symbolov a a b v slove ostal zachovaný), preto $w \in L$.

- $S \to bSa$: $S \underset{G}{\Rightarrow} bSa \overset{*}{\underset{G}{\Rightarrow}} bw'a = w \qquad |w'| = n,\ |w| = n+2$
  Opäť podľa indukčného predpokladu $w' \in L$. Slovo $w = bw'a$ spĺňa podmienky príslušnosti do jazyka L, teda $w \in L$.

- $S \to SS$: $S \underset{G}{\Rightarrow} SS \overset{*}{\underset{G}{\Rightarrow}} w_1 w_2$ $\quad w_1 \in \{aw_1'b,\ bw_1'a,\ \varepsilon\},\ w_2 \in \{aw_2'b,\ bw_2'a,\ \varepsilon\}$
  $|w_1'| = n_1,\ |w_2'| = n_2$

  V tomto prípade pravidlo negeneruje priamo terminálny symbol, preto nijako neovplyvňuje pomer počtu výskytov symbolu a a b v slove. Podľa indukčného predpokladu $w_1', w_2' \in L$ a ako sme ukázali v predošlých krokoch, platí: $aw_1'b$, $bw_1'a$, $aw_2'b$, $bw_2'a$, $\varepsilon \in L$. Týmto pravidlom sa tieto slová môžu reťaziť za seba, čo neporušuje podmienky príslušnosti do jazyka.

Ukázali sme, že pre každý prípad reťazca w dĺžky $n+2$ generovaného gramatikou G, platí $w \in L$. Indukčný krok platí.

(ii) $\mathbf{L \subseteq L(G)}$

Potrebujeme ukázať, že všetky slová z jazyka L je možné generovať gramatikou G. Pre dôkaz použijeme indukciu vzhľadom k dĺžke slova w:

Bázový prípad: $|w| = 0$, slovo $w = \varepsilon \in L$ a zároveň existuje pravidlo $S \to \varepsilon$ v gramatike G. Z toho vyplýva že $\varepsilon \in L(G)$.

Indukčný predpoklad: Predpokladajme, že $L \subseteq L(G)$ platí pre všetky slová dĺžky $n-2$, formálne:
$\forall w' \Big(w' \in L \wedge |w'| = n-2 \implies S \overset{*}{\underset{G}{\Rightarrow}} w' \Big)$

Indukčný krok: Ukážeme, že implikácia platí aj pre slová dĺžky n
tj. že platí: $\forall w \Big(w \in L \wedge |w| = n \implies S \overset{*}{\underset{G}{\Rightarrow}} w \Big)$

Podľa definície jazyka L existuje niekoľko prípadov slova $w \in L$ takého, že $|w| = n$:

- $w = aw'b$ $\quad |w'| = n-2,\ |w| = n$

  Podľa indukčného predpokladu platí $S \overset{*}{\underset{G}{\Rightarrow}} w'$. Pri derivovaní slova w z gramatiky G budeme postupovať nasledovne: $S \underset{G}{\Rightarrow} aSb \overset{*}{\underset{G}{\Rightarrow}} aw'b$. Platí $w \in L(G)$.

- $w = bw'a$ $\quad |w'| = n-2,\ |w| = n$

  Analogicky ku predošlému prípadu, použijeme indukčný predpoklad a derivácia slova w z gramatiky G bude vyzerať nasledovne: $S \underset{G}{\Rightarrow} bSa \overset{*}{\underset{G}{\Rightarrow}} bw'a$.
  Platí $w \in L(G)$.

- $w = axa$ $\quad |x| = n-2,\ |w| = n$

  Pre x zároveň platí: $x \in \{a,b\}^* \wedge \#_a(x) = \#_b(x) - 2$.
  Reťazec x rozpíšeme do tvaru $x_1, x_2, \ldots, x_{n-2}$. Postupne vytvoríme reťazce w_i pre každé $i \in \langle 1, n \rangle$ tak, že w_i je prefixom slova w dĺžky i, tj.:
  $w_1 = a$,
  $w_2 = ax_1$,
  $w_3 = ax_1x_2$,
  $\ldots$
  $w_{n-1} = ax_1x_2 \ldots x_{n-2}$,
  $w_n = ax_1x_2 \ldots x_{n-2}a$,

  Všimnime si, že platia nasledovné vzťahy:

$\#_a(w_1) = \#_b(w_1) + 1$	slovo w_1 obsahuje o jedno a **viac** ako b
$\#_a(w_{n-1}) = \#_b(w_{n-1}) - 1$	slovo w_{n-1} obsahuje o jedno a **menej** ako b

Z toho vyplýva, že musí existovať j také, pre ktoré platí:

$$1 < j < n - 1 \quad \wedge \quad \#_a(w_j) = \#_b(w_j)$$

a teda aj: $w_j \in L$.

Nech $w' = w_j$. Slovo w tak môžeme zapísať ako $w = w'w''$, keďže w' je prefixom w. Ďalej je zrejmé, že ak v celom slove w, aj v jeho prefixe w' je pomer zastúpenia symbolov a a b rovnaký, platí to aj pre w'', formálne:

$$w = w'w'' \wedge w, w' \in L \implies w'' \in L$$

Podľa indukčného predpokladu $S \overset{*}{\underset{G}{\Rightarrow}} w'$ a $S \overset{*}{\underset{G}{\Rightarrow}} w''$. Pre deriváciu slova w z gramatiky G postupujeme nasledovne: $S \underset{G}{\Rightarrow} SS \overset{*}{\underset{G}{\Rightarrow}} w'S \overset{*}{\underset{G}{\Rightarrow}} w'w'' = w$. Platí $w \in L(G)$.

- pre prípad $w = bxb$ je postup analogický k $w = axa$, rovnako platí $w \in L(G)$.

Ukázali sme, že pre každý prípad reťazca w z jazyka L dĺžky n platí $w \in L(G)$. Indukčný krok platí.

Pomocou indukcie k dĺžke slova $w \in L$ sme dokázali, že výrok: $L(G) \subseteq L \ \wedge \ L \subseteq L(G)$ je pravdivý, z čoho plynie, že je pravdivý aj výrok: $L = L(G)$.

2. Uvažujte *doprava čítaný jazyk* TS M, značený ako $L^P(M)$, ktorý je definovaný ako množina reťazcov, ktoré M príjme v behu, pri ktorom nikdy nepohne hlavou *doľava* a nikdy neprepíše žiadny symbol na páske za iný. Dokážte, či je problém prázdnosti doprava čítaného jazyka TS M, tj. či $L^P(M) = \emptyset$, je rozhodnuteľný:
   - ak *áno*, napíšte algoritmus v pseudokóde, ktorý daný problém bude rozhodovať;
   - ak *nie*, dokážte nerozhodnuteľnosť redukciou z jazyka *HP*.

Riešenie

Základná myšlienka algoritmu pre rozhodovanie tohto problému je postupný prevod TS M na deterministický konečný automat A_D so zreteľom na fakt, že jazyk, ktorý automat A_D prijíma nieje jazykom $L(M)$ ale $L^P(M)$, čo predstavuje jeho podmnožinu.

Algorithm 1 Pseudokód algoritmu, ktorý rozhoduje problém prázdnosti doprava čítaného jazyka TS M, tj. či $L^P(M) = \emptyset$.

Vstup: Turingov stroj M = $(Q, \Sigma, \Gamma, \delta, q_0, q_F)$
Výstup: TRUE, ak je jazyk $L^P(M)$ prázdny, inak FALSE

1: Vytvoríme TS $M' = (Q, \Sigma, \Gamma, \delta', q_0, q_F)$, kde
$\delta' : (Q \setminus \{q_F\}) \times \Gamma \to Q \times (\Gamma \cup \{R\})$ taká, že:
$\forall q_1, q_2 \in Q \; \forall a \in \Gamma :$
$(q_2, R) \in \delta'(q_1, a) \iff (q_2, R) \in \delta(q_1, a)$
$(q_2, a) \in \delta'(q_1, a) \iff (q_2, a) \in \delta(q_1, a)$

2: Zavedieme relácie $\xrightarrow{\Delta}$ a $\xrightarrow{\Delta R}$ v množine Q takto:
$\forall q_1, q_2 \in Q : q_1 \xrightarrow{\Delta} q_2 \overset{def}{\iff} (q_2, \Delta) \in \delta(q_1, \Delta)$,
$\forall q_1, q_2 \in Q : q_1 \xrightarrow{\Delta R} q_2 \overset{def}{\iff} (q_2, \Delta) \in \delta(q_1, \Delta) \vee (q_2, R) \in \delta(q_1, \Delta)$,
kde Q je konečná množina vnútorných stavov nejakého TS a δ jeho prechodová funkcia.

3: Následne definujeme funkcie :
Δ-uzáver$(p) = \{q \in Q \mid p \xrightarrow{\Delta}{}^{*} q\}$.
ΔR-uzáver$(p) = \{q \in Q \mid p \xrightarrow{\Delta R}{}^{*} q\}$.

4: Pre výpočet Δ-uzáveru (resp. ΔR-uzáveru) potom môžeme použiť Warshallov algoritmus a postupovať analogicky ako pri výpočte ε-uzáveru pri RKA.

5: Prevedieme TS M' na RKA $A = (Q, \Sigma_A, \delta_A, q_0, F)$ nasledujúcim postupom:
$\Sigma_A = \Sigma \cup \{\varepsilon\}$
$F = \{q_F\}$
$\delta_A : Q \times \Sigma_A \to 2^Q$ taká, že:
$\forall q_1, q_2 \in Q \; \forall a \in \Sigma_A \setminus \{\varepsilon\} :$
$q_2 \in \delta_A(q_1, a) \iff (q_2, R) \in \delta'(q_1, a)$
$q_2 \in \delta_A(q_1, \varepsilon) \iff (q_2, a) \in \delta'(q_1, a) \wedge \big((q_2, a) \neq \emptyset \vee (q_2 = q_F)\big)$
$\forall q_1, q_2 \in Q :$
$q_2 \in \delta_A(q_1, \varepsilon) \iff (q_2, \Delta) \in \delta'(q_1, \Delta) \wedge q_2 \in \Delta$-uzáver$(q_0)$
$q_2 \in \delta_A(q_1, \varepsilon) \iff (q_2, R) \in \delta'(q_1, \Delta) \wedge q_1 \in \Delta$-uzáver$(q_0)$
$q_2 \in \delta_A(q_1, \varepsilon) \iff (q_2, \Delta) \in \delta'(q_1, \Delta) \wedge q_F \in \Delta R$-uzáver$(q_2)$
$q_2 \in \delta_A(q_1, \varepsilon) \iff (q_2, R) \in \delta'(q_1, \Delta) \wedge q_F \in \Delta R$-uzáver$(q_2)$

6: Pomocou algoritmu z prednášky prevedieme RKA A na DKA $A_D = (Q_D, \Sigma_A, \delta_D, q_{0D}, F_D)$.

7: Ak platí: $\exists q \in Q_D : (q \in F_D \wedge q$ je dostupný z $q_{0D})$ vráť **FALSE**, inak vráť **TRUE**.

Dôvodom tohto prístupu k riešeniu je, že jazyk $L^P(M)$ vieme prijímať pomocou tzv. read-only right moving TS, keďže pre prijatie slova nieje povolený prepis symbolu na páske za iný a hlavou je povolené posúvať iba doprava. Navyše read-only TS dokáže prijímať iba regulárne jazyky. Môžeme tak využiť faktu, že v triede regulárnych jazykov $\mathcal{L}_3$ je problém neprázdnosti rozhodnuteľný.

V prvom kroku vytvoríme TS M', ktorý predstavuje TS M bez pravidiel pri ktorom sa hlava pohybuje doľava alebo prepisuje symbol na páske za nejaký iný. Ďalej si potrebujeme zadefinovať funkcie Δ-uzáver a ΔR-uzáver, ktoré nám pomôžu pre definovanie špeciálnych prípadov pri prepise prechodovej funkcie TS M' na prechodovú funkciu rozšíreného konečného automatu. Pri prepise je potrebné ošetriť niekoľko špeciálnych prípadov. Ak by v TS došlo k zaseknutiu, v automate k nemu dôjde takisto. Ďalej je potrebné riešiť náhradu prechodov tvaru Δ/Δ na začiatku pred čítaním vstupného reťazca, tvaru Δ/R práve pred začiatkom čítania slova a prechodov tvaru Δ/Δ alebo Δ/R po prečítaní reťazca na ceste do koncového stavu za ε-prechody. Vzniknutý RKA A prevedieme na DKA A_D, u ktorého vieme rozhodnúť, či jazyk, ktorý prijíma, je prázdny.

3. Uvažujte jazyk $L_{42} = \{\langle M \rangle \mid \text{TS } M \text{ zastaví na niektorom vstupe tak, že páska bude obsahovať práve 42 neblankových symbolov}\}$. Dokážte pomocou redukcie, že L_{42} je nerozhodnuteľný. Uveďte ideu dôkazu čiastočnej rozhodnuteľnosti L_{42}.

Riešenie

Dôkaz, že $\mathbf{L_{42}}$ je nerozhodnuteľný

Pre dôkaz nerozhodnuteľnosti použijeme techniku redukcie z problému A na problém B, zapisujeme: $A \leq B$. Potrebujeme ukázať, že problém daný jazykom L_{42} je nerozhodnuteľný, preto v zápise redukcie L_{42} odpovedá B problému a za problém A zvolíme problém zastavenia (HP), o ktorom vieme, že je nerozhodnuteľný.

Redukcia $\mathbf{HP} \leq \mathbf{L_{42}}$:

- Problém **HP** je charakterizovaný jazykom:
  $HP = \{\langle M \rangle \# \langle w \rangle \mid M \text{ je TS, ktorý na reťazci } w \text{ zastaví.}\}$
- Problém $\mathbf{L_{42}}$ je charakterizovaný jazykom:
  $L_{42} = \{\langle M \rangle \mid M$ je TS, ktorý zastaví na niektorom vstupe tak, že páska bude obsahovať práve 42 neblankových symbolov$\}$.
- Navrhneme redukciu $\sigma : \{0, 1, \#\}^* \to \{0, 1\}^*$ z jazyka HP na L_{42}.
- σ priradí každému reťazcu $x \in \{0, 1, \#\}$ reťazec $\langle M_X \rangle \in \{0, 1\}^*$, kde M_X je TS, ktorý so vstupným reťazcom $w \in \{0, 1\}^*$ pracuje nasledovne:
  (i) M_X zmaže svoj vstup w zo vstupnej pásky.
  (ii) M_X zapíše na vstupnú pásku reťazec x, ktorý má uložený v konečnom stavovom riadení.
  (iii) M_X overí, či x má štruktúru $x_1 \# x_2$, kde x_1 je kód TS a x_2 je kód jeho vstupu. Ak nie, **zmaže** svoju vstupnú pásku a **odmietne**.
  (iv) M_X odsimuluje na reťazci s kódom x_2 beh TS s kódom x_1. Ak simulácia skončí, TS M_X zmaže svoju vstupnú pásku, **zapíše** na vstupnú pásku 42 symbolov a, pričom platí $a \neq \Delta$ a **príjme**, inak **cyklí**.
- Funkciu σ je možné implementovať úplným TS M_σ, ktorý pre vstup x vyprodukuje kód TS M_X. Ten sa skladá zo štyroch komponent odpovedajúcim vyššie uvedeným krokom:
  (i) M_σ vypíše kód pre TS, ktorý zmaže vstupnú pásku;
  (ii) M_σ vypíše kód takého TS, ktorý zapíše na vstup reťazec $x = x_1, x_2, \ldots, x_n$ (tento TS zapíše na vstup x_1, posunie hlavu doprava, zapíše na vstup x_2, ...);
  (iii) M_σ vypíše kód TS, ktorý na vstupe overí, či sa jedná o platnú inštanciu HP a ak nie tak M_X a **odmietne**;
  (iv) M_σ vypíše kód TS, ktorý spustí univerzálny TS; UTS simuluje TS s kódom x_1 a vstupom x_2.

  Komponenty z bodov (i),(iii),(iv) možno pripraviť dopredu, pretože nijak nezávisia na vstupe x a M_σ len vypíše príslušný kód. M_σ ešte potrebuje zaistiť sekvenčné predávanie riadenia medzi týmito komponentami.

- Skúmajme možné jazyky TS M_X:
  (a) $L(M_X) = \emptyset \iff$ (x nemá správnu štruktúru) alebo (x má správnu štruktúru $x_1\#x_2$, ale TS s kódom x_1 na vstupe s kódom x_2 cyklí).
  (b) $L(M_X) = \Sigma^* \iff$ (x má správnu štruktúru) a zároveň (TS s kódom x_1 na vstupe s kódom x_2 zastaví).
- Ukážeme, že funkcia redukcie σ zachováva členstvo v jazyku:
  $\forall x \in \{0,1,\#\}^* :$ $\sigma(x) = \langle M \rangle \in L_{42} \iff L(M_X) = \Sigma^* \iff$ (x má štruktúru $x_1\#x_2$, kde x_1 je kód TS a x_2 je taký kód vstupu, že TS s kódom x_1 zastaví na vstupe s kódom x_2) $\iff x \in HP$.

Dôkaz čiastočnej rozhodnuteľnosti L_{42} (idea)

K čiastočnému rozhodnutiu problému L_{42} môžeme zostrojiť TS M', ktorý na svojej páske simuluje beh TS $M \in L_{42}$ pre jednotlivé vstupné reťazce. Avšak TS M' nemôže vygenerovať vstupné reťazce a postupne na každom spustiť neobmedzenú simuláciu, pretože môže nastať prípad, že sa na nejakom reťazci zacyklí a k simulácii ďalšieho reťazca už nedôjde. Nebolo by tak garantované, že nájdeme reťazec, ktorý TS M príjme (ak taký existuje). Namiesto toho, M' na svojej páske postupne rozbieha viacero simulácií TS M pre jednotlivé vygenerované reťazce. Tie potrebuje vhodným spôsobom oddeliť a pri každej simulácii si ešte naviac pamätať, aký je aktuálny stav riadenia pri spracovaní daného vstupu. Simulácia potom prebieha nasledovne:

(1) TS M' prejde všetky aktuálne rozbehnuté simulácie a na každej urobí jeden krok.
(2) Ak na niektorej zo simulácii TS M príjme príslušný reťazec, TS M' príjme.
(3) Inak pridá nový reťazec pre simuláciu na pásku spolu so stavom riadenia (počiatočnú konfiguráciu) a postup opakuje.

Je zrejmé, že M' príjme, ak $L(M) \neq \emptyset$. Inak neskončí.