Skúška z predmetu TZI, riadny termín, 7.1.2008
FIIT STU, Mgr. Daniela Chudá, PhD.

Nasledovné riešenia ukazujú návod ako postupovať pri riešení zadaní, ide o jedno z možných riešení zadaní.

1. **Aké stromové prehľadávanie** je nutné použiť pri prehľadávaní stromu konfigurácií pri dôkaze tvrdenia, že **L(DTM) = L(NTM)**. Odpoveď krátko zdôvodnite. *(3body)*

Prehľadávanie do šírky, pretože v strome konfigurácií môže existovať nekonečne dlhá vetva.

2. **Je každá bezkontextová gramatika kontextovou?** Odpoveď krátko zdôvodnite. *(3 body)*

NIE ≈ ÁNO S VÝNIMKOU gramatík obsahujúcich ε pravidlá. Bezkontextová gramatika môže obsahovať ε pravidlá, kontextová nie, pretože jej pravidlá sú neskracovacie (viď def. 2.2.2).

3. Definícia nedeterministického zásobníkového automatu je NPDA=(K, Σ, Γ, δ, q_0,Z_0,F), kde konfigurácia je trojica (q,w,γ) ∈ Kx Σ^*x Γ^*. **Definujte krok výpočtu NPDA.** *(6 bodov)*

Krok výpočtu nedeterministického zásobníkového automatu *je relácia* $\vdash_A$ *na množine konfigurácií* $K \times \Sigma^* \times \Gamma^*$ *definovaná nasledovne:*

$$(q, aw, Z\gamma) \vdash_A (p, w, \beta\gamma), \qquad ak \qquad (p, \beta) \in \delta(q, a, Z),$$

pričom $p, q \in K$, $a \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$, $w \in \Sigma^*$, $\beta, \gamma \in \Gamma^*$.

4. Nad triedou jazykov *L* je daná ternárna operácia (tj. s troma argumentmi) ⊗, definovaná tak, že pre ľubovoľné jazyky L_1, L_2, L_3 platí: ⊗(L_1, L_2, L_3) = $(L_1 \cdot L_2)^* \cup L_3$. Zistite a zdôvodnite, **či je trieda bezkontextových jazykov L_{CF} uzavretá vzhľadom na operáciu ⊗. Ak existuje konštruktívny dôkaz, uveďte ho!** *(8 bodov)*

JEDNO Z MOŽNÝCH RIEŠENÍ:

- uvedenie predpokladov konštruktívneho dôkazu
Nech L1, L2, L3 sú ľubovoľné bezkontextové jazyky.
Potom **musia existovať bezkontextové** gramatiky G1, G2, G3, ktoré ich generujú. Formálne:
Bezkontextová g. G1=(N1,T1,P1,S1) generuje jazyk L1, teda L(G1)=L1,
" " G2=(N2,T2,P2,S2) generuje jazyk L2, teda L(G2)=L2,
" " G3=(N3,T3,P3,S3) generuje jazyk L3, teda L(G3)=L3.

Predpokladajme, že: $N1 \cap (N2 \cup T2) = \emptyset$, $N1 \cap (N3 \cup T3) = \emptyset$, $N2 \cap (N1 \cup T1) = \emptyset$,
$N2 \cap (N3 \cup T3) = \emptyset$, $N3 \cap (N1 \cup T1) = \emptyset$, $N3 \cap (N2 \cup T2) = \emptyset$.
Ak by tieto predpoklady neplatili, je potrebné vhodne premenovať neterminálne symboly.

-konštrukcia konštruktívneho dôkazu
Zostrojíme novú gramatiku G_{new} nasledujúcim spôsobom.
Nech G_{new} = (N1∪N2∪N3∪{S_0,X}, T1∪T2∪T3, P1∪P2∪P3∪P', S_0)

Nová množina pravidiel P' je vytvorená tak, že do nej patria nasledujúce pravidlá:
$S_0 \rightarrow X \mid S3$
$X \rightarrow S1.S2.X \mid \varepsilon$

- záver konštruktívneho dôkazu
Z uvedenej konštrukcie vyplýva, že gramatika G_{new} generuje jazyk (L1.L2)* ∪L3 teda, že:
$L(G_{new}) = (L1.L2)^* \cup L3$.
Z definície bezkontextovej gramatiky vyplýva, že všetky pravidlá v P' sú **bezkontextové**.

Skúška z predmetu TZI, riadny termín, 7.1.2008
FIIT STU, Mgr. Daniela Chudá, PhD.

Podľa **predpokladu** tiež platí, že všetky pravidlá v P1, P2, P3 sú **bezkontextové**.
Z toho vyplýva, že aj množina P1∪P2∪P3∪P' obsahuje iba bezkontextové pravidlá.
Preto aj jazyk $L(G_{new})$ musí byť bezkontextový. Tým je vrdenie je dokázané.

5. V registroch R1, R2 sú uložené dve nezáporné celé čísla x a y. Napíšte počítadlový stroj, ktorý realizuje výpočet funkcie $f(x,y)=(x+2y)^2$ nedeštruktívne a výsledok ukladá do registra R3. *(10 bodov)*

$(s_1a_4a_6a_7)_1(s_2a_5a_6a_6a_7a_7)_2(s_4a_1)_4(s_5a_2)_5(s_6(s_7a_3a_8)_7(s_8a_7)_8)_6$
vysvetlenie:
R_1: x, R_2: y R_3:výsledok $f(x,y)=(x+2y)^2$
R_4: x, R_5: y R_6:x+2y R_7:x+2y
vrátenie x,y do R_1, R_2
umocnenie na druhú $R_6 * R_7$

$(s_1a_4a_6a_7)_1(s_2a_5a_6a_6a_7a_7)_2$
$(s_4a_1)_4(s_5a_2)_5$
$(s_6(s_7a_3a_8)_7(s_8a_7)_8)_6$

6. **Dokážte, že X-DOWHILE-IF-AbstractMachine+ je ekvivalentný s počítadlovým strojom (Abacus Machines).** (Dokazujú sa dve implikácie s využitím rekurzívnej definície AM.)
X-DOWHILE-IF-AbstractMachine+ : Je daná množina registrov R0 (jediný akumulátor), R1, ..., v ktorých je možné reprezentovať nezáporné celé čísla. Je daný počítač X-DOWHILE-IF-AM+, ktorý používa nasledujúce inštrukcie počítača RAM:

LOAD i	*(prekopíruje obsah registra Ri do akumulátora)*	
STORE i	*(prekopíruje obsah akumuátora do registra Ri)*	
ADD =i	*ADD i*	*(inštrukcia realizuje operáciu sčítania)*
SUB =i	*SUB i*	*(inštrukcia realizuje operáciu odčítania Θ , čiže napr. 5 Θ 10 = 0)*

a namiesto skokov používa konštrukciu *do { ... } while(GZERO)* ,
čo je štandardný dowhile cyklus z C-jazyka, ktorý vykonáva svoje telo minimálne raz, opakuje sa pokiaľ hodnota akumulátora R0 je kladná (vačšia ako nula).

Používa tiež konštrukciu if(GZERO){ ... }

čo je štandardný príkaz if BEZ else-vetvy z C-jazyka, ktorý vykoná svoje telo ak platí, že obsah akumulátora je kladný, teda R0 > 0.

Vstupné argumenty pre X-DOWHILE-AM+ sa nachádzajú pred výpočtom v registroch, (nemá inštrukcie READ a WRITE). *(20 bodov)*

Pri dokazovaní ekvivalencie **X-DOWHILE-IF-AbstractMachine+ a Abacus Machines** musíme dokázať obe implikácie:

1. **X-DOWHILE-IF-AbstractMachine+ ⇨ Abacus Machines**
Mapovanie registrov identické $R_i \rightarrow R_i$

LOAD i
STORE i
ADD =i
SUB =i
ADD i
SUB i
do {X} while (GZERO)
if (GZERO) {…}

↕ prepísať do inštrukcií AM

2. **X-DOWHILE-IF-AbstractMachine+ ⇦ Abacus Machines**
Mapovanie registrov posunuté
$R_i \rightarrow R_{i+1}$ (reg AM: $R_i \rightarrow$ reg. X-DOWHILE-IF-AbstractMachine+ R_{i+1})

a_i	LOAD i+1
	ADD =1
	STORE i+1

Skúška z predmetu TZI, riadny termín, 7.1.2008
FIIT STU, Mgr. Daniela Chudá, PhD.

s_i

```
LOAD i+1
SUB =1
STORE i+1
```

postupnosť M_1M_2 M_3... M_Z

predpokladám, že mám M_1,M_2 ,M_3... ,M_Z

indukčný predpoklad M_i~P_i

M_1M_2 M_3... M_Z skonštruujem tak, že inštrukcie budú za sebou P_1P_2 P_3... P_Z

$(M)_i$

```
LOAD i+1
if (GZERO) {do {P load i+1}while(GZERO) }
```