

## ÚVOD DO TI 2022 - ÚLOHA 2

### Všeobecné poznámky

1. Iba výsledok nestačí. Treba písať aj zdôvodnenia. Ak nájdete niečo na internete, uveďte aspoň zdroj, ideálne sa to snažte dokázať a vysvetliť.

### Poznámky k príkladu 1

(opravoval Michal Winczer)

Hodnotenie častí a) 3 body, b) 4 body, c) 3 body  
Najčastejšie chyby:

- V a) resp b) ste využívali platnosť c), ktorú ste ale nedokázali. –1b.
- Ukázali ste len jednu inklúziu. Polovica bodov.
- Pokus o kontrapríklad. 0.5b
- Pokus o dôkaz bol za 0.5 až 1b

### poznámky k príkladu 2

(opravovala Dana Pardubská)

Príklad bol za 10b (2,2,3,3).

Formálny dôkaz nerobil skoro nik.

Chyby:

- definíciu Kleeneho uzáveru nenapísal skoro nik, nestrhávala som body, keďže bolo jasné, že to bolo na cvikách. Len vtedy, keď ani nenapísali, čomu sa  $L^*$  rovná (ako začiatok nekonečnej množiny slov) Niekde bolo, že v tej zátvorke to bolo len po  $a^n$ , neuvedomujú si, že vtedy je to vlastne konečný jazyk
- ak napísali a minimálne jednou vetou odôvodnili, prečo sú oba jazyky slova párnej dĺžky, je to za plný počet bodov
- Nevedia zapísať  $\cup_{1 \leq i \leq n}$  čosi, ale píšú to len ako jednu zátvorku s  $2n - 1$  (1 bod dole), v lepšom prípade k tomu na riadok napíšu  $n \in N$  (0.5 bodu dole)
- Asi odniekiaľ opisovali, lebo veľmi veľa malo značenie  $L_1, L_2$  namiesto  $L_a, L_b$  (nechcelo sa mi za to strhávať body). Viacerí to ale veľmi pekne odôvodnili, ako k tomu zápisu prišli

## Poznámky k príkladu 3

(opravoval Michal Winczer)

Na tento príklad si trúfalo len zopár z vás. Zohľadnili sme, že také niečo ste asi ešte nikdy neriešili.

- Testovanie na rôznych vstupoch nie je dôkaz správnosti programu. Ak nájdete pri testovaní chybu, máte istotu, že program nie je správny. Ak nenájdete chybu nedozviete sa nič o správnosti programu!
- Pri dokazovaní správnosti programu je treba sformulovať tvrdenie, ktoré dokazujete. Je možné, že tých tvrdení bude viac. Napríklad keď program vykoná riadok č.  $x$ , bude platiť tvrdenie  $T_x$ . Nemusí to byť pre každý riadok programu, ale napríklad pred začiatkom, alebo po skončení cyklu, na konci then, resp. else vetvy v podmienenom príkaze, na konci funkcie a pod.
- Počet operácií, ktorý program musí na každom vstupe vykonať sa zvyčajne dá ľahšie určiť v najlepšom prípade, resp. v najhoršom prípade. Samozrejme ak je jasné, ktorý prípad je najlepší resp. najhorší. Priemerný počet operácií, alebo očakávaný počet, t.j. koľko v priemere program vykoná operácií na náhodnom vstupe, je zvyčajne podstatne ťažšie. Tu sme sa zaujímali o najhorší možný prípad. Keď nevieme presvedčivo zistiť, ktorý prípad je najhorší, môžeme na to ísť okľukou a nájsť nejaký výraz (obsahujúci premenné nášho programu), a všimnúť si ako sa mení jeho hodnota. Tu sme mali vnorený cyklus, teda nebolo očividné, že celkovo sa nevykonajú príkazy vo vnútornom cykle viac než je  $c|x|$ , kde  $x$  je vstupný reťazec. Pri troche šťastia (s pomôckou so zadania) sa dalo všimnúť, že sa hodnota výrazu pri prechode cyklom zmenší, pričom nikdy nebude záporná. Teda max. počet je max. hodnota výrazu pre daný vstup. Takouto technikou (klesajúci parameter) sa dá ukázať o programe aj to, že skončí, pre každý vstup. To sa tiež niekedy hodí.