

## ÚVOD DO TI 2017 - ÚLOHA 3

### Všeobecné poznámky

1. Nezabúdajte na papier napísať deň, kedy chodíte na cvičenie alebo meno cvičiacej, aby sa k vám jednoduchšie dostali úlohy.
2. Ak máte jeden príklad na viac papierov, zopnite ich, nech sa nám nestratia a neprídete kvôli tomu o body.
3. Iba výsledok nestačí. Treba písať aj zdôvodnenia. Ak nájdete niečo na internete, uveďte aspoň zdroj, ideálne sa to snažte dokázať a vysvetliť.
4. Výsledky nemusíte vyčíslovať, stačí zapísanie pomocou sumy. Ak však používate nejaké označenie namiesto vzorcov, napíšte aspoň všeobecný tvar.

(opravovala Lucia Budinská) Obe časti boli za 0.5b.

- a) Uznávala som aj neformálne dôkazy, pokiaľ ste poriadne vysvetlili, prečo sa dané jazyky rovnajú. Body som strhávala, ak ste niečo zabudli vysvetliť (pri neformálnom dôkaze), mali zvláštne odvodenie (pri formálnom dôkaze), alebo nejaké chyby. Ak ste riešili dôkaz pomocou matematickej indukcie, vhodné bolo ukazovať, že slovo dĺžky  $k$  patrí do jedného aj druhého jazyka, a nie že  $k$ -ta mocnina jedného jazyka sa rovná  $k$ -tej mocnine druhého (to, že sa rovnajú Kleeneho uzávery, neznamená, že sa rovnajú aj jazyky - viď. predchádzajúca úloha).
- b) Plný počet ste dostali, ak ste našli konkrétne slovo, ktoré do jedného jazyka patrilo a do druhého nie, a vysvetlili prečo; alebo ak ste vysvetlili, aké sú rozdiely medzi jazykmi. V prípade, že vám niečo chýbalo, resp. ste mali nejaké drobné chyby, strhávala som 0.5-1b.

### Poznámky k príkladu 1.2

(opravovala Paula Budzáková)

Väčšina ste mali príklad správne. Až na pár jedincov, ktorý zle pochopili zadanie a dokazovali každé jedno tvrdenie zo zadania. V takomto prípade máte za príklad 0 bodov.

- To, či je jazyk regulárny ste mali dokázať pomocou matematickej indukcie. Boli prípady, kde ste len vysvetlili ideu, ktorá aj keď bola správna chýbal vám dôkaz. V takomto prípade ste mali -5b.
- Bázu ste mali ukázať pre  $k = 0$ ,  $k = 1$  a  $k = 2$  (podobne, ako v minulej úlohe). Ak ste len odpísali tvrdenia zo zadania, alebo napísali, že to triviálne platí a podobne - 2b. Ak ste mali len pre  $k = 0$  -1,5b. Ak vám chýbalo  $k = 2$  - 1b. Bol to jednoduchý príklad a už z minulej úlohy som si všimla, že akosi bázu poriešite ukázaním pre nulovú dĺžku slova a to je všetko. Veľakrát ale tým neukážete absolútne nič. Bola by som rada, aby ste si zapamätali, že ak budete ešte niekedy dokazovať nejaký príklad pomocou matematickej indukcie, báza musí obsahovať všetky tieto tri hodnoty.

- Ak niekomu chýbal indukčný predpoklad - 1b
- A ak ho niekto aj mal, ale jeho indukčný predpoklad vyzeral ako  $L^n - 0,5b$ . Myšlienky zatiaľ čítať neviem.

## poznámky k príkladu 1.3

(m.w)

- Pre zadané slovo  $w$ , vygenerujete všetky slová dĺžky  $|w|$  nad  $abc1$ . Tých ale môže byť ozaj veľa:  $|abc1|^{|w|}$ . Pre každé jedno z nich skúšať či sa nezobrazí na  $w$ , teda je možné, ale prakticky to bude fungovať len pre krátke slová a malé abecedy. -3b
- Vygenerujete všetky možnosti dĺžok podreťazcov slova  $w$ , ktorých je tiež veľa a tie sa snažíme "pokryť" daným homomorfizmom. Ak sa to podarí pridáme do výsledku. -3b
- Backtrackingom vytvárame slová, ktoré sú potenciálny vzor slova  $w$ . Využívame na to zadaný homomorfizmus  $h$ . Nešikovne sa to dá spraviť tak, že vzor vytvárame pokým jeho obraz nemá dĺžku  $|w|$  bez toho aby sme vznikajúci obraz porovnávali so slovom  $w$ . Lepšie je samozrejme s backtrackovaním prestať keď vznikajúci obraz nie je prefix  $w$  -2b
- Ako predchádzajúci bod, ale rôzne viac alebo menej úspešné optimalizácie.
  - Vzory sa usilujeme vytvoriť iba z tých znakov, ktoré sa zobrazia na znaky, ktoré sa nachádzajú v slove  $w$
- Rôzne drobné chyby
- Chýbajúce akékoľvek vysvetlenie hlavnej myšlienky riešenia. Pomôže aj uviesť čo je v ktorej premennej - napr. najmenší prvok poľa  $z$  a doteraz prezretých prvkov a pod.
- program dá zlý výsledok. Žiadne vysvetlenie hlavnej myšlienky. -8b

Ukážka programov, ktorý všetkým odporúčame preštudovať!

(Zoltán Onódy)

```
def vzor(abc1, abc2, homomorphism, word):
    originals = []
    def f(word, ans):
        if not word:
            originals.append(ans)
        for frm, to in homomorphism:
            if word.startswith(to):
                f(word[len(to):], ans + frm)
    f(word, '')
    return originals
```

(Roman Kuruc)

```
def vzor(abc1, abc2, h, w):
    pole = []
    for i, y in h:
        if y in w:
            if w.index(y) == 0:
                if len(w) > len(y):
                    for z in vzor(abc1, abc2, h, w[len(y):len(w)+1]):
                        pole.append(i+z)
            else:
                pole.append(i)
    return pole
```