Domáca úloha 1

Lucia Korbeľová

Dôkaz správnosti programu P3

Program P3 vyzerá nasledovne:

Program P3

initially r = 0

assign r := max{ f(r), g(r), h(r) }

end{P3}

V nasledujúcej úlohe dokážeme, že program P3 naozaj korektne nájde najbližší vyhovujúci čas pre tri osoby, kedy sa môžu stretnúť.

1. Najprv dokážeme, že neurobí nič zlého, teda nepreskočí žiaden voľný spoločný termín. To môžeme dokázať sporom. Predpokladajme, že termín z je najbližší možný termín, kedy sa všetci traja môžu stretnúť. A predpokladajme, že sa po nejakom kroku priradenia do r v programe stane, že $r > z$. To by sa stalo v prípade, že v tomto kroku programu, kedy dostaneme $r > z$, musela jedna z funkcií f, g alebo h vrátiť hodnotu, ktorá je väčšia ako z. Avšak to je v spore s tvrdením, že z je najbližší voľný termín. A to že z má byť v skutočnosti najbližší voľný termín spoločného stretnutia znamená, že všetci traja majú vtedy čas, preto sa nemohlo stať, že jedna z funkcii f, g alebo h by vrátila že $r > z$. Z definície týchto funkcií vieme, že každá z nich pre ľubovoľné r vráti ako nasledujúcu hodnotu $f(r), g(r), h(r)$, ktorá je väčšia alebo rovná hodnote r a predstavuje najbližší možný termín, kedy ma daná osoba čas na stretnutie po termíne r. Teda nemohlo by sa stať, že vrátila termín po z, pretože najprv by musela každá funkcia vrátiť hodnotu z. A tu sme sa dostali k sporu a teda sme dokázali, že program P3 určite nepreskočí žiaden voľný spoločný termín.
2. Následne dokážeme, že program P3 „urobí nakoniec niečo dobre“, teda nájde spoločný termín. Z predchádzajúceho dôkazu vieme, že nikdy nemôže platiť, že $r > z$, čiže určite P3 nepreskočí spoločný voľný termín stretnutia. Takže pre r bude vždy platiť: $r \leq z$. Ak $r = z$, tak sa nám podarilo nájsť najbližší spoločný termín a naše tvrdenie platí. Ak však $r < z$, tak vieme že aspoň jedna z funkcií f, g alebo h v nasledujúcom priradení assign r := max{ f(r), g(r), h(r) }, vráti hodnotu, ktorá je väčšia ako r. Teda napríklad f(r) > r. Z toho vyplýva, že sa aj hodnota premennej r v tomto kroku bude musieť zväčšiť. Z toho môžeme usúdiť, že v ľubovoľnom kroku vykonávania programu P3 sa hodnota r zväčší oproti predchádzajúcemu kroku výpočtu až dokým nebude platiť, že $r = z$. Teda musí sa stať, že po nejakom počte krokov program nájde spoločný termín.

Z dôkazov 1. a 2. plynie, že termín nájdený programom P3 bude prvý možný termín.

Dôkaz správnosti programu sort1

Program sort1 vyzerá nasledovne:

Program sort1

assign $\langle \mathbb{I}\ j: 0 \leq j < N :: A[j], A[j+1] \coloneqq A[j+1], A[j]\ if\ A[j] > A[j+1] \rangle$

end{sort1}

Dokážeme, že program sort1 funguje a korektne utriedi pole A.

1. Najprv dokážeme, že program v žiadnom kroku nezhorší stav utriedenia oproti predchádzajúcemu kroku, teda, že sa počet nesprávne utriedených dvojíc v poli ($i < j \ \wedge A[i] > A[j]$) nikdy nezväčší.

   Vieme, že v každom kroku môžu nastať dve možné prípady – buď sa nevymenia hodnoty na políčkach $A[j], A[j+1]$ alebo sa vymenia. Ak sa hodnoty v jednotlivých políčkach polia nevymenili, pretože už bola táto dvojica dobre utriedená, tak určite nestúpol počet nesprávne utriedených dvojíc, keďže nenastala žiadna zmena v poli. Čiže počet ostane rovnaký ako v predchádzajúcom kroku.

   Ak nastala druhá možnosť a hodnoty na políčkach $A[j], A[j+1]$ sa vymenili, keďže tato dvojica bola nesprávne utriedená, tak sa určite nezväčšil počet nesprávne utriedených dvojíc v poli. To vyplýva z toho, že sme vymenili hodnoty v týchto dvoch susedných políčkach, takže táto susedská dvojica je od tohto momentu správne utriedená. Čiže táto jedna nesprávne utriedená dvojica z poľa zmizla. Taktiež určite nepribudne žiadna nová nesprávne utriedená dvojica medzi $A[j]$ a $A[i]$ (pre $i \neq j, i \neq j+1$), keďže výmenou hodnôt na políčkach $A[j], A[j+1]$ sa nezmenilo poradie medzi $A[j]$ a $A[i]$, čiže nijako sa vzťah medzi nimi nezmenil napriek tomu, že hodnota v $A[j]$ sa predtým nachádzala v $A[j+1]$. Tým pádom nemohli pribudnúť žiadne nove nezrovnalosti medzi $A[j]$ a žiadnym iným políčkom. Analogicky si vieme to isté odvodiť pre políčko $A[j+1]$. Z toho vyplýva, že celkový počet nesprávne utriedených dvojíc sa po výmene susedných políčok $A[j], A[j+1]$ musel zmenšiť o jedna.

   Z tohto vieme usúdiť, že v žiadnom prípade sa počas behu programu nemôže udiať to, že by program utriedenie poľa zhoršil oproti predchádzajúcemu kroku. Každý krok buď zanechá pole bez zmeny alebo zmenší počet nesprávne utriedených dvojíc.

2. Následne dokážeme, že program „urobí nakoniec niečo dobré“, teda, že určite po nejakom počte krokov programu sort1 bude pole A správne utriedené (počet nesprávne utriedených dvojíc sa bude rovnať 0). Toto tvrdenie nám priamo vyplýva z dôkazu číslo 1, ktorým sme ukázali, že počet nesprávne zoradených dvojíc v poli sa nemôže zväčšovať. Buď po vykonaní jedného kroku programu sort1 bude počet nezrovnalostí rovnaký alebo o niečo menší. A nakoľko predpokladáme, že sa príkaz priradenia assign $\langle \mathbb{I}\ j: 0 \leq j < N :: A[j], A[j+1] \coloneqq A[j+1], A[j]\ if\ A[j] > A[j+1] \rangle$ bude vykonávať nekonečne veľa krát, musí nastať moment, kedy sa počet

nezrovnalostí zníži až na nula. Čiže určite program po neznámom počte krokov korektne utriedi prvky poľa A.

Pomocou dôkazov 1. a 2. sa nám podarilo dokázať, že program sort1 funguje správne a utriedi pole A.