

PROGRAMOVANIE PARALÉLNYCH A DISTRIBUOVANÝCH SYSTÉMOV

DOMÁCA ÚLOHA 2

Autor: Marián Kravec

Úloha 1

Majme program binomial:

```
Program binomial
assign
   $\langle \square n: 0 \leq n < N ::$ 
     $c[n, 0] := 1 \ \square c[n, n] := 1$ 
     $\square \langle \square k: 0 < k < n ::$ 
       $c[n, k] := c[n-1, k-1] + c[n-1, k]$ 
     $\square$ 
end{binomial}
```

Podme ukázať, že tento program robí to čo od neho očakávame. Najprv si vytvoríme metriku, ako bolo spomenuté na prednáške, vhodnou metriku môže byť v tomto prípade počet správne vyplnených vrstiev od vrchu plus zlomok správne vyplnených políčok prvej neúplne správnej vrstvy. Túto metriku chceme maximalizovať. Keďže máme N vrstiev (konečný počet) a každá vrstva má konečný počet políčok tak aj počet vrstiev aj zlomok správne vyplnených políčok v neúplnej vrstve zhora ohraničený s konečným počtom hodnôt. Vďaka tomu ak dokážeme, že tento algoritmus túto metriku nikdy nezníži a občas zvýši, dokážeme, že algoritmus v nekonečnom čase určite správne vyplní binomický trojuholník (maticu c).

- Najprv ukážeme, že tento algoritmus metriku nikdy nezníži, čiže neurobí nič zlé - dokážeme to sporom, na to aby metrika klesla algoritmus by musel zmeniť správnu hodnotu v bunke na niektorú z prvých už správnych vrstiev (alebo na prvej neúplnej) na nesprávnu - ak je bunka na okraji vrstvy (druhú súradnicu ná 0 alebo n) algoritmu na danú pozíciu dá 1 čo je správna hodnota, ak bunky nie je okrajová, tak spočíta hodnoty na vyššej vrstve a keďže predpokladáme, že všetky vyššie vrstvy sú správne dostávame sa k sporu a preto môžeme tvrdiť, že algoritmus neurobí nič zlé
- Teraz chceme ukázať, že aspoň občas urobí niečo dobré, inými slovami zvýši metriku ak nie je už dosiahnuté maximum metriky. Metriku vieme zvýšiť tým, že na prvej nie úplne správne vyplnenej vrstvy doplníme správnu hodnotu (respektíve keď vyplníme celú prvú nie úplne správne vyplnenú vrstvu čo je iba špeciálny prípad) - keďže existuje príkaz pre vyplnenie každej bunky tak existuje aj príkaz na vyplnenie nesprávne vyplnenej bunky na prvej nie úplne správne vyplnenej vrstve, ak je táto bunka okrajová, zmení jej hodnotu na 1 čiže nová hodnota bude správna, ak okrajová nie je sčíta hodnoty vrstvy nad touto vrstvou o ktorej predpokladáme, že je správne vyplnená, čiže aj v tejto bunke bude nová hodnota správna a tým pádom sa zvýši hodnota metriky, čo je to čo sme chceli dokázať. Špeciálnym prípadom je prvá vrstva (teoreticky aj druhá) ktorá nemá nad sebou žiadnu vrstvu avšak dôkaz platí aj pre ňu keďže jediná bunka na nej je okrajová z čoho vyplýva, že existuje príkaz ktorý ju správne vyplní aj napriek absencii vrstvy nad ňou
- Nakoniec chceme ukázať, že program dosiahne pevný bod kde je celý binomický trojuholník správne vyplnený. Tu využijeme už ukázané/dokázané vlastnosti, ukázali sme, že hodnota metriky neklesne, a že vždy existuje príkaz ktorý ju zvýši. Takisto sme na začiatku ukázali, že vďaka konečnému počtu vrstiev a buniek má táto metrika konečný počet hodnôt a existuje maximum, z tohto vyplýva, že vďaka tomu, že vždy kým nie je metrika maximálna existuje príkaz ktorý ju zvýši (a žiaden iný ju nezníži) a každý príkaz sa vykoná nekonečne veľa krát tak metrika dosiahne v konečnom čase maximum čo je v pre tento program pevný bod. Maximum metriky je, že sú všetky riadky binomického trojuholníku (matice) správne vyplnené, čiže v pevnom bode máme kompletne správne vyplnený binomický trojuholník

Rozdiel medzi tým, keď začíname s maticou inicializovanou s nulovými a jednotkovými hodnotami veľmi malý, keďže dôkaz nepredpokladá nič o počiatočných hodnotách v bunkách, teoreticky, ak by boli v bunkách hodnoty 1 tak sú prvé dve vrstvy už vopred správne vyplnené... ale neviem či to nejak pomáha dôkazu.

Úloha 2

- a) $\{x > 3\}x := x + 5\{x > 4\}$ - pravdivé - ak platí $x > 3$ tak po $x := x + 5$ bude platiť $x + 5 > 4$, keďže to vieme upraviť na $x > -1$ a $x > 3$ implikuje, že $x > -1$
- b) $\{x > 3\}x := x + 5\{x > 2\}$ - pravdivé - ak platí $x > 3$ tak po $x := x + 5$ bude platiť $x + 5 > 2$, keďže to vieme upraviť na $x > -3$ a $x > 3$ implikuje, že $x > -1$
- c) $\{x > 3\}x := x + 5\{y > 2\}$ - nepravdivé - poznáme podmienku pre premennú x a príkaz ktorý ju zmení čo nám nehovorí nič o premennej y
- d) $\{x > 3\}x := x + 5\{x > 6\}$ - pravdivé - ak platí $x > 3$ tak po $x := x + 5$ bude platiť $x + 5 > 6$, keďže to vieme upraviť na $x > 1$ a $x > 3$ implikuje, že $x > -1$
- e) $\{\text{true}\}x := 2.x\{x \text{ je párne}\}$ - nepravdivé - ak nevieme či je x kladné celé číslo tak nevieme povedať či bude eventually párne
- f) $\{\text{false}\}x := 2.x\{x \text{ je párne}\}$ - pravdivé - tvrdenia začínajúce false sú vždy pravdivé (ak som správne pochopil prednášku)
- g) $\{k \text{ je celé kladné číslo}\}x := k.x\{x \text{ je zložené číslo}\}$ - nepravdivé, ak nevieme, či x je celé kladné číslo toto tvrdenie nemusí platiť

Úloha 3

O programe Binomial môžeme sformulovať napríklad takéto vlastnosti:

- $c[n, 0] \neq 1 \wedge n \geq 0 \wedge n \leq N$ ensures $c[n, 0] = 1$ - iba jeden príkaz dokáže zmeniť hodnotu $c[n, 0]$ a ten to zmení iba na 1
- $c[0, 0] = 1$ stable - v podstate len špecifický prípad predchádzajúceho tvrdenia avšak ak sú hodnoty iniciované na 1 tak platí: $c[n, 0] = 1$ invariant
- $c[n, k] \neq \binom{n}{k} \wedge n \geq 0 \wedge n \leq N \wedge k \leq n$ leads-to $c[n, k] = \binom{n}{k}$ - v podstate samotná podstata programu, hovoriaca, že eventually bude v každej bunke správna hodnoty binomického koeficientu