Polosemestrálna skúška: IB015 14.11. 2008 17.00

Bodovanie

2 príklady s možnosťami A-E: správna odpoveď 8b, chybná -2b, žiadna 0b
dôkaz indukciou – 0b - 16b

Príklad 1

Výstup 4 z možností je [[0], [1,0], [2,1,0], [3,2,1,0], [4,3,2,1,0],[5,4,3,2,1,0], ...], jedna možnosť má iný výstup. Ktorá? (zadane boli definície (nie typové anotácie) funkcii: `map`, `iterate`, `zipWith`)

```
A ..........[[k, k-1..0] | k<-[0..]]
B ..........iterate (\t -> map (1+) t++[0]) [0]
C ..........let s = [0]:map (zipWith (+) [1,1..]) s in s
D ..........iterate (\t -> (head t+1):t) [0]
E ..........let s = [0] : map (\t -> (1+head t):t) s in s
```

Príklad 2

Zadaných bolo 5 „rovníc“ (zložená funkcia = iná funkcia). Bolo potrebné určiť, ktorá z rovností je platná. (bolo zadane „(f.g) x = f (g x)“ a „operátor (.) skladá iba *unárne* funkcie“)

```
A ..........odd.(2*) = even
B ..........(*).(*)  = (^2)
C ..........(^).(2*) = \x y -> 2*(x^y)
D ..........(*).(1+) = \x y -> y + x*y
E ..........(&&).(||)= \x y z -> x || (y && z)
```

Príklad 3

Dokážte indukciou rovnosť (definície (nie typové anotácie) funkcií `filter` a `negate` (ostatné v 1. príklade)):

```
map negate (filter (<0) s) = filter (>0) (map negate s)
```

Riešenie na druhej strane nemusí byť správne. Podľa počtu bodov z písomky neskôr sa pokúsim opraviť chyby

Príklad 1 – riešenie

Stačí si rozpísať možnosti podľa definícií a výsledok bude jasný ☺.

Iný výstup má možnosť C. Ak skopírujete toto do hugsu, uvidíte tvar prvých 10 položiek zoznamu:

```
let s = [0] : map (zipWith (+) [1,1..]) s in take 10 s
```

Príklad 2 – riešenie

Ako je napísané v zadaní, operátor (.) je iba pre unárne funkcie. Teda ak ho aplikujeme na binárnu f-ci, musíme doplniť aj nejaký parameter. Správna je možnosť D:

```
(*).(1+)            = \x y -> y + x*y
((*).(1+)) a b      = (\x y -> y + x*y) a b  --L: f.g x = f(g x)
(*)((1+) a) b       = (\x y -> y + x*y) a b
(*)(1+a) b          = (\x y -> y + x*y) a b  --P: x=a, y=b
(1+a)*b             = (b + a*b)
b + a*b             = b + a*b
```

Príklad 3 – riešenie

Indukciu budeme viesť cez zoznam s, postupujeme podľa definícií funkcií `map` a `filter`.

Báza: s=[] --prázdny zoznam

```
map negate (filter (<0) []) = filter (>0) (map negate [])
map negate ([])             = filter (>0) ([])
[]                          = []
```

Indukčný predpoklad:

```
map negate (filter (<0) s) = filter (>0) (map negate s)
```

Indukčný krok: pre zoznam (x:s) ... teda o jeden prvok (x) dlhší

```
IP=>map negate (filter (<0) (x:s))=filter (>0) (map negate (x:s))

map negate (filter (<0) (x:s))   =  filter (>0) (map negate (x:s))
```

riešenie rozdelíme na 2 časti: x>=0 a x<0

x>=0 **-- ak x>=0 tak (-x)<=0**

```
map negate (filter (<0) (x:s))   =  filter (>0) (map negate (x:s))
           -----------------             ----------------
-- if x<0 then x:t else t             -- negate x: map negate s
-- where t = filter (<0) s            -- (-x):map negate s
--- ^to je DEF filter
--- pre x>=0 vráti filter (<0) s
map negate (filter (<0) s)       =  filter (>0) ((-x):map negate s)
                                    --DEF filter ˇ
                                    -- if ((-x)>0) ... neplatí

map negate (filter (<0) s)       =  filter (>0) (map negate s)
-- toto je indukčný predpoklad – tvrdenie platí pre x>=0
```

x<0 **-- ak x<0 tak (-x)>0**

```
map negate (filter (<0) (x:s))   =  filter (>0) (map negate (x:s))
-- if x<0 then x:t else t             -- negate x: map negate s
-- where t = filter (<0) s            -- (-x):map negate s
--- ^to je DEF filter
--- pre x>=0 x:filter (<0) s
map negate (x:filter (<0) s)     =  filter (>0) ((-x):map negate s)
- negate x:map negate (filter (<0) s)    --DEF filter ˇ
                                    -- if ((-x)>0) ... platí

(-x):map negate (filter (<0) s)  =  (-x):filter (>0) (map negate s)
-- toto je indukčný predpoklad rozšírený na oboch stranách o (–x), takže
-- tvrdenie opäť platí
```