

nov 06, 05 19:00

dp05a-zh1-mlmegol.txt

Page 1/4

Deklaratív programozás nagyvárthelyi
Budapest, 2005. október 27.
=====

SML megoldások, V1.1, dp05s-zh1-mlmegol.txt

A csoport

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két statikus szemantikai hiba van. Melyek ezek?

- (a) `[op<(1, 2.3), false, "bb" = #"b"]`
- real és int nem hasonlítható össze
- string és char nem hasonlítható össze
- (b) `(chr 127, 2+4 = 4*4, ~1) = ("B", 2444, ~(5-7))`
- char és string nem hasonlítható össze
- bool és int nem hasonlítható össze
- (c) `foldl op^ ~1.0 [(1,0),(3,4),(2,1)]`
- `op^` helyett más operátornak kellene lennie,
pl. `(fn ((x,y),z) => x*y+z)`
- a második argumentumnak itt int-nek kellene lennie

Pontozás (összesen max. 7 pont):

5.a - 5.c Helyes válasz 2-2-3 pont.
Minden hiba megtalálása 1 pontot ér, kivéve az 5.c-t, ahol a nem megfelelő operátor használatának felismerésére 2 pontot adunk.

6. Mi a `t` értéke és típusa az alábbi, egymástól független deklarációk kiértékelése után?

- (a) `val (::_:~t::_) = explode "X" @ tl(explode "mas")`
`t = #"s" : char`
- (b) `val (::_:~t) = List.filter Char.isDigit (explode "alb2c3d4")`
`t = [#"3", #"4"] : char list`
- (c) `val t = map (fn (a,b) => a>b)`
`[(4+0,2*2), (1,2), (size "ab", size "bc")]`
`t = [false, false, false] : bool list`

Pontozás (összesen max. 7 pont):

6.a - 6.c Helyes válasz (1+1)-(1+1)-(2+1) pont -- (érték+típus)

7. Legyen

```
fun foldb f e 0 = e | foldb f e n = foldb f (f(n,e)) (n-1)
```

Mi az `x` értéke az alábbi, egymástól független deklarációk kiértékelése után?

- (a) `val x = foldb (fn (i, k) => i+2+k) 0 0`
`x = 0`
- (b) `val x = foldb (fn (i, k) => i*2+k) 1 1`
`x = 3`
- (c) `val x = foldb (fn (i, k) => i*2*k) 1 3`
`x = 48`

nov 06, 05 19:00

dp05a-zh1-mlmegol.txt

Page 2/4

```
(d) val x = map (foldb (fn (i, k) => i*i+k) 0) [1,2,3]
      x = [1,5,14]
```

```
(e) val x = List.filter (fn i => i mod 3 <> 0)
      (map (foldb (fn (i, k) => i+k) 0) [1,2,3,4,5])
      x = [1,10]
```

Pontozás (összesen max. 8 pont):

7.a-7.b-re helyes válaszért 1-1, 7.c-7.e-re 2-2 pont jár.

8. Leszállónak nevezzük az olyan (a,b,c) hármast, amelyre $a-b > b-c$, leszállási meredekségének pedig az $(a-b)-(b-c)$ különbséget.

Írjon olyan függvényt `leszm` néven, amelynek egy hármásokból álló egészlista az argumentuma, az eredménye pedig az argumentumlistában található leszálló hármások leszállási meredekségének tetszőleges sorrendű listája.

Lehetőleg magasabbrendű függvény(ek) alkalmazásával oldja meg a feladatot. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.

```
(* leszm: (int * int * int) list -> int list
   leszm ns = az ns listában lévő leszálló hármások leszállási
   meredekségének tetszőleges sorrendű listája
*)
```

Példák:

```
leszm [] = [];
leszm [(6,4,3)] = [1];
leszm [(6,4,2)] = [];
leszm [(6,4,0),(6,4,1),(6,4,2),(6,4,3),(6,4,4),(9,5,2),(9,5,1)] = [1,2,1];
```

Egy megvalósítása magasabbrendű függvénnyel:

```
fun leszm ns = map (fn (a,b,c) => (a-b)-(b-c))
  (List.filter (fn (a,b,c) => a-b>b-c) ns);
```

Pontozás (összesen max. 8 pont):

Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont levonás. Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy a végtelen rekurzió (-3 pont). Segédfüggvényben a fejkomment hiánya: -2 pont, de nem követelmény a fejkommentben a függvény típusának specifikálása. Ha nagyon rossz a program hatékonysága, -2 pont a levonás.

nov 06, 05 19:00	dp05a-zh1-mlmegol.txt	Page 3/4
Deklaratív programozás nagyzárthelyi Budapest, 2005. október 27. ===== SML megoldások, V1.1, dp05s-zh1-mlmegol.txt		

B csoport		

5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben kifejezésenként két-két statikus szemantikai hiba van. Melyek ezek?		
(a) [1.0 = 2, op>("a", "b"), ord #"a" mod 3] - real és int nem hasonlítható össze - az ord #"a" mod 3 nem bool típusú a listában		
(b) (chr 64, 4, ~12) = (ord #"b", 2+2 = (4 div 1), ~(7-7)) - a chr 64 nem int típusú - a 2+2 = 4 div 1 nem int típusú		
(c) foldr (op/) [1.4, 3.4, 2.1] ~10 - foldr paramétereinek sorrendje hibás (f xs e), helyesen: f e xs - ~10 típusa int, real típusú érték kell helyette		
Pontozás (összesen max. 7 pont): 5.a - 5.c Helyes válasz 2-2-3 pont. Minden hiba megtalálása 1 pontot ér, kivéve az 5.c-t, ahol a nem megfelelő argumentumsorrend felismerésére 2 pontot adunk.		
6. Mi a t értéke és típusa az alábbi, egymástól független deklarációk kiértékelése után?		
(a) val (_::_:t::_) = tl(explode "ban") @ explode "g" t = #"g" : char		
(b) val (_::_:_:t) = List.filter Char.isAlpha (explode "12abc34d") t = ["#d"] : char list		
(c) val t = map (fn (a,b) => a>=b) [(2,1), (2+2,4*2), (ord #"B",ord #"A")] t = [true, false, true] : bool list		
Pontozás (összesen max. 7 pont): 6.a - 6.c Helyes válasz (1+1)-(1+1)-(2+1) pont -- (érték+típus)		
7. Legyen		
<pre>fun foldj f e 0 = e foldj f e n = f(foldj f e (n-1), n)</pre>		
Mi az x értéke az alábbi, egymástól független deklarációk kiértékelése után?		
(a) val x = foldj (fn (i, k) => i+3+k) 0 0 x = 0		
(b) val x = foldj (fn (i, k) => i*3+k) 1 1 x = 4		
(c) val x = foldj (fn (i, k) => i*3*k) 1 2 x = 18		
(d) val x = map (foldj (fn (i, k) => i*i+k) 0) [3,2,1] x = [12,3,1]		

nov 06, 05 19:00	dp05a-zh1-mlmegol.txt	Page 4/4
<pre>(e) val x = List.filter (fn i => i mod 3 <> 0) (map (foldj (fn (i, k) => i+k) 0) [5,4,3,2,1]) x = [10,1]</pre>		
Pontozás (összesen max. 8 pont): 7.a-7.b-re helyes válaszáért 1-1, 7.c-7.e-re 2-2 pont jár.		
Megjegyzés: foldj paraméterezése a feladatban eltér a szokásostól. Az előadáson ez szerepelt:		
<pre>fun foldj f e 0 = e foldj f e n = f(n, foldj f e (n-1)),</pre>		
a feladatban pedig ez:		
<pre>fun foldj f e 0 = e foldj f e n = f(foldj f e (n-1), n)</pre>		
vagyis a különbség az f argumentumainak sorrendjében van.		
Nem vonunk le pontot, de megjelöljük, ha a (d) kérdésben a feladatnak megfelelő [12,3,1] válasz helyett a szokásos működés szerinti [14,5,1] a válasz. A többi esetben mindkét változatnál ugyanaz a válasz.		
8. Elszállónak nevezzük az olyan (a,b,c) hármast, amelyre a-b<b-c, elszállási meredekségének pedig a (b-c)-(a-b) különbséget.		
Írjon olyan függvényt elszm néven, amelynek egy hármastól álló egészlista az argumentuma, az eredménye pedig az argumentumlistában található elszálló hármastok elszállási meredekségének tetszőleges sorrendű listája.		
Lehetőleg magasabbrendű függvény(ek) alkalmazásával oldja meg a feladatot. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.		
<pre>(* elszm: (int * int * int) list -> int list elszm ns = az ns listában lévő elszálló hármastok elszállási meredekségének tetszőleges sorrendű listája *)</pre>		
Példák:		
<pre>elszm [] = []; elszm [(5,4,2)] = [1]; elszm [(5,4,3)] = []; elszm [(5,4,0),(5,4,1),(5,4,2),(5,4,3),(5,4,4),(7,5,3),(7,5,2)] = [3,2,1,1];</pre>		
Egy megvalósítása magasabbrendű függvénnyel:		
<pre>fun elszm ns = map (fn (a,b,c) => (b-c)-(a-b)) (List.filter (fn (a,b,c) => a-b<b-c) ns);</pre>		
Pontozás (összesen max. 8 pont): Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont levonás. Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy a végtelen rekurzió (-3 pont). Segédfüggvényben a fejkomment hiánya: -2 pont, de nem követelmény a fejkommentben a függvény típusának specifikálása. Ha nagyon rossz a program hatékonysága, -2 pont a levonás.		