

```

1 Deklaratív programozás nagyábrathelyi
2 Budapest, 2006. április 21.
3 =====
4 SML megoldások, V1.0, dp06s-zh1-mlmegol.txt
5 -----
6 A csoport
7 -----
8
9
10 5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben
11 kifejezésenként két-két statikus szemantikai hiba van. Melyek ezek?
12
13 (a) [op>{"a", "b"}, (1, 2) <> (1, 2, 3), true = false]
14 - pár ((1,2)) és hármas ((1,2,3)) nem hasonlítható össze
15 - string ("b") és char (#"a") nem hasonlítható össze
16
17 (b) (2*3 = 3+3, chr 95, ~9) = (6*1, "b", 0-5-4)
18 - bool (2*3=3+3) és int (6*1) nem hasonlítható össze
19 - char (chr 95) és string ("b") nem hasonlítható össze
20
21 (c) foldl ~1 [4, 2, 6, 4, 1, 2] op/
22 - az op/ művelet nem alkalmazható int-ekre
23 - foldl paraméterezése rossz; helyesen: (op div) ~1 [4, 2, 6, 4, 1, 2]
24
25 Pontozás (összesen max. 7 pont):
26 5.a - 5.c: helyes válasz 2-2-3 pont.
27 Minden hiba megtalálása 1 pontot ér, kivéve az 5.c-t, ahol a nem
28 megfelelő operátor használatának felismerésére 2 pontot adunk.
29
30 6. Mi a q értéke az alábbi, egymástól független deklarációk kiértékelése
31 után?
32
33 (a) val (_:::q:_) = explode "hús" @ rev ["t", "#é", "#v"]
34 q = #"s"
35
36 (b) val (_:::q) = List.filter Char.isAlpha (explode "4a3r2ald")
37 q = ["a", "#d"]
38
39 (c) val q = map (fn (a,b) => a>b) [(7,2*3), (1,2), (ord #"A",ord #"Z")]
40 q = [true, false, false]
41
42 Pontozás (összesen max. 7 pont):
43 6.a - 6.c: helyes válasz 2-2-3 pont. Hibákért 1-1 pont levonás.
44
45 7. Tekintsük a következő függvénydefiníciókat!
46
47 (* sf : 'a list * 'a list -> 'a list *)
48 fun sf (ms, ns) = foldl op:: ns ms
49 (* fl : ('a * 'a) list -> 'a list *)
50 (* f2 : ('a * 'b) list * 'a list * 'b list -> 'a list * 'b list *)
51 fun fl mns = let fun f2 ((i,j)::ijs, ms, ns) = f2(ijs, i::ms, j::ns)
52 | f2 ([], ms, ns) = (ms, ns)
53 in sf(f2(mns, [], []))
54 end
55
56 Mi az x értéke az alábbi, egymástól független deklarációk kiértékelése
57 után?
58
59 (a) sf ([0, 2, 4], [1, 3, 5])
60 x = [4, 2, 0, 1, 3, 5]
61
62 (b) sf ([1.0, 2.0, 3.0], [0.0, 0.0])
63 x = [3.0, 2.0, 1.0, 0.0, 0.0]
64
65 (c) fl [(0,1), (2,3), (4,5)]
66 x = [0, 2, 4, 5, 3, 1]
67
68 (d) fl [{"a","A"}, {"b","B"}, {"c","C"}]
69 x = [{"a", "B"}, {"c", "C"}, {"B", "A"}]
70
71

```

```

72 (e) foldl op^ "" (fl [{"n",""}, {"o",""}, {"p",""}])
73 x = "pon"
74
75 Pontozás (összesen max. 8 pont):
76 7.a-7.b-re helyes válaszért 1-1, 7.c-7.e-re 2-2 pont jár.
77 Ha foldl-t ***következésként*** foldr-nek értelmezi, 7.a-7.b-re
78 nem, de 7.c-7.e-re megkaphatja a pontokat.
79
80 Megjegyzések:
81 sf (ms, ns) = List.revAppend(ms, ns) = rev ms @ ns
82 fl mns = az mns-beli párok első tagjaiból képzett lista
83 foldl op^ "" ts = rev ts
84
85 8. Tekintsük az alábbi adattípus-deklarációt:
86
87 datatype 'a D = X of 'a D list | Y of 'a
88
89 Emelkedőnek nevezzük az olyan (x, y, z) hármasokat, amelyekre x < y < z.
90 Írjon olyan függvényt emelkedő néven, amely egy (int * int * int) D
91 típusú adatstruktúrában található emelkedő hármasok számát adja
92 eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény
93 alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.
94
95 (* emelkedok : (int * int * int) D -> int
96 emelkedok t = a t-beli emelkedő hármasok száma *)
97
98 Példák:
99
100 emelkedok(X[]) = 0;
101 emelkedok(Y(3,4,4)) = 0;
102 emelkedok(Y(3,4,5)) = 1;
103 emelkedok(X[X[],X[],Y(3,4,5)]) = 1;
104 emelkedok(X[X[Y(2,5,8),Y(6,9)],X[X[Y(8,9)],Y(-2,3,8)],X[],Y(3,4,5)]) = 3;
105
106 Egy megoldás foldl-lal:
107
108 fun emelkedok (Y(a,b,c)) = if a < b andalso b < c then 1 else 0
109 | emelkedok (X ns) = foldl (fn (a, b) => emelkedok a + b) 0 ns
110
111 Hasonló megoldás map-pel és foldl-lal:
112
113 fun emelkedok (Y(x,y,z)) = if x < y andalso y < z then 1 else 0
114 | emelkedok (X xs) = foldl op+ 0 (map emelkedok xs)
115
116 Pontozás (összesen max. 8 pont):
117 Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont
118 levonás. Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy
119 a végtelen rekurzio (-3 pont). Segédfüggvényben a fejkomment hiánya: -2
120 pont, de nem követelmény a fejkommentben a függvény típusának
121 specifikálása. Ha nagyon rossz a program hatékonysága, -2 pont a
122 levonás.
123
124

```

```

125 Deklaratív programozás nagyábrathelyi
126 Budapest, 2006. április 21.
127 =====
128 SML megoldások, V1.0, dp06s-zh1-mlmegol.txt
129 -----
130 B csoport
131 -----
132 -----
133
134 5. Az alábbi, egymástól független, szintaktikailag helyes SML-kiifejezésekben
135 kiifejezésenként két-két statikus szemantikai hiba van. Melyek ezek?
136
137 (a) [2 < 2.3, "a" > # "b", 7 mod 3 = 0]
138 - int (2) és real (2.3) nem hasonlítható össze
139 - string ("a") és char (#"b") nem hasonlítható össze
140
141 (b) (64.0, 1*4, ~12) = (ord #"b", 2+2 = 4 div 1, 0-7-7)
142 - real (64.0) és int (ord #"b") nem hasonlítható össze
143 - int (1*4) és bool (2+2 = 4 div 1) nem hasonlítható össze
144
145 (c) foldr (op div) ~10 [1.4, 0.6, 3.4, 4.7, 2.1, 1.9]
146 - a div művelet nem alkalmazható real-ekre
147 - foldr második argumentuma (~10) int, itt real típusú kellene
148
149 Pontozás (összesen max. 7 pont):
150 5.a - 5.c: helyes válasz 2-2-3 pont.
151 Minden hiba megtalálása 1 pontot ér, kivéve az 5.c-t, ahol a nem
152 megfelelő operátor használatának felismerésére 2 pontot adunk.
153
154 6. Mi a p értéke az alábbi, egymástól független deklarációk kiértékelése
155 után?
156
157 (a) val (_,_:):p::_ = tl [# "v", #"a", #"t"] @ explode "ta"
158 p = #"a"
159
160 (b) val (_,_:):p::_ = List.filter Char.isLower (explode "ld2a3r4a")
161 p = [# "a"]
162
163 (c) val p = map (fn (a,b) => a<b) [(4+2,7), (2,1), (ord #"9",ord #"8")]
164 p = [true, false, false]
165
166 Pontozás (összesen max. 7 pont):
167 6.a - 6.c: helyes válasz 2-2-3 pont. Hibákért 1-1 pont levonás.
168
169 7. Tekintsük a következő függvénydefiníciókat!
170
171 (* sg : 'a list * 'a list -> 'a list *)
172 fun sg (ms, ns) = foldr op:: ns ms
173 (* g1 : ('a * 'a) list -> 'a list *)
174 (* g2 : 'a list * 'b list * ('a * 'b) list -> 'b list * 'a list *)
175 fun g1 mns = let fun g2 (ms, ns, (i, j)::ijs) = g2(j::ms, i::ns, ijs)
176 | g2 (ms, ns, []) = (ns, ms)
177 in
178   sg(g2([], [], mns))
179 end
180
181 Mi az x értéke az alábbi, egymástól független deklarációk kiértékelése
182 után?
183
184 (a) val x = sg ([0, 2, 4], [1, 3, 5])
185 x = [0, 2, 4, 1, 3, 5]
186
187 (b) val x = sg ([1.0, 2.0, 3.0], [0.0, 0.0])
188 x = [1.0, 2.0, 3.0, 0.0, 0.0]
189
190 (c) val x = g1 [(0,1), (2,3), (4,5)]
191 x = [4, 2, 0, 5, 3, 1]
192
193 (d) val x = g1 [("a","A"), ("b","B"), ("c","C")]
194 x = ["c", "b", "a", "C", "B", "A"]
195

```

```

196 (e) val x = foldr op^ "" (gl [{"n",""}, ("o",""), ("p","")])
197 x = "pon"
198
199 Pontozás (összesen max. 8 pont):
200 7.a-7.b-re helyes válaszért 1-1, 7.c-7.e-re 2-2 pont jár.
201 Ha foldr-t ***következésként*** foldl-nek értelmezi, 7.a-7.b-re
202 nem, de 7.c-7.e-re megkaphatja a pontokat.
203
204 Megjegyzések:
205 sg (ms, ns) = ms @ ns
206 gl mns = az mns-beli párok első tagjaiból képzett lista fordítottja
207 foldr op^ "" ts = ts
208
209 8. Tekintsük az alábbi adattípus-deklarációt:
210
211 datatype 'a H = U of 'a H list | V of 'a
212
213 Súlyedőnek nevezünk az olyan (x, y, z) hármastokat, amelyekre x > y > z.
214 Írjon olyan függvényt sullyedok néven, amely egy (int * int * int) H
215 típusú adatstruktúrában található sullyedó hármastok számát adja
216 eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény
217 alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.
218
219 (* sullyedok : (int * int * int) H -> int
220 sullyedok t = a t-beli sullyedó hármastok száma *)
221
222 Példák:
223
224 sullyedok(U[]) = 0;
225 sullyedok(V(3,3,4)) = 0;
226 sullyedok(V(5,4,3)) = 1;
227 sullyedok(U[U[],U[],V(5,4,3)]) = 1;
228 sullyedok(U[U[V(8,5,2),V(6,6,9)],U[V(8,7,7)],V(8,3,-2)]),U[,V(5,4,3)]) = 3;
229
230 Egy megoldás foldl-lal:
231
232 fun sullyedok (V(a,b,c)) = if a > b andalso b > c then 1 else 0
233 | sullyedok (U hs) = foldl (fn (a, b) => sullyedok a + b) 0 hs
234
235 Hasonló megoldás map-pel és foldl-lal:
236
237 fun sullyedok (V(x,y,z)) = if x > y andalso y > z then 1 else 0
238 | sullyedok (U xs) = foldl op+ 0 (map sullyedok xs)
239
240 Pontozás (összesen max. 8 pont):
241 Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont
242 levonás. Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy
243 a végtelen rekurzió (-3 pont). Segédfüggvényben a fejkomment hiánya: -2
244 pont, de nem követelmény a fejkommentben a függvény típusának
245 specifikálása. Ha nagyon rossz a program hatékonysága, -2 pont a
246 levonás.
247
248

```