

```

1 Deklaratív programozás nagyábrhelyi
2 Budapest, 2006. május 11.
3 =====
4 SML megoldások, V1.0, dp06s-zh2-mlmegol.txt
5
6 -----
7 A csoport
8 -----
9
10 5. Az alábbi, egymástól független, szintaktikailag helyes SML-kiifejezésekben
11 kiifejezésenként két-két statikus szemantikai hiba van. Melyek ezek?
12
13 (a) [#a = "b", (1, 2) < (2, 1), 1 > 2 = true]
14 - char (#"a") és string ("b") nem hasonlítható össze
15 - párok - (1,2)és (2,1) - kisebb-nagyobb volta nem vizsgálható
16
17 (b) (3+3, chr 93.0, 7) = (3*2, #"b", 0-3-4, 0.0)
18 - hármás és négyes nem hasonlítható össze
19 - chr-nek nem lehet real típusú argumentuma (93.0)
20
21 (c) List.filter [4, 2, 6, 4, 1, 2] (op div)
22 - a paraméterek sorrendje helyes: List.filter p xs
23 - p helyén predikátumnak kellene lennie, (op div) nem az
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 "sin" @ rev ["t", #"é", #"r"]
34 q = ["#r", #"é", #"t"]
35
36 (b) val (_::q:_) = List.filter Char.isAlpha (explode "4a3r2ald")
37 q = #"r"
38
39 (c) val q = #2(foldl (fn (x,(y,b)) => (x,b andalso xy)) (9,true) [3,2,1])
40 q = true
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 : string list * string -> string *)
48 (* f1 : string list * string list -> string list -> string list *)
49 (* f2 : string list * string list -> string -> string *)
50 fun sf (ms, ns) = foldl op^ ns ms
51 and f1 (m::ms, n::ns) rs = f1 (ms, ns) (m^n::rs)
52 | f1 _rs = rs
53 and f2 mens t = sf(f1 mns [], t)
54
55 Mi az x értéke az alábbi, egymástól független deklarációk kiértékelése
56 után?
57
58 (a) val x = sf (["a", "n", "pa"], "ma")
59 x = "panama"
60
61 (b) val x = f1 (["a", "b", "c"], [",", ""]) []
62 x = ["b", "a"]
63
64 (c) val x = f2 (["a", "b", "c"], [",", ""]) "d"
65 x = "abd"
66
67 (d) val x = f2 (["k", "k", "k"], ["u", "a"]) "c"
68 x = "kukac"
69
70 (e) val x = let val g = map str o explode in f2 (g "eee", g "smlk") "" end
71 x = "esemel"

```

```

72 Pontozás (összesen max. 8 pont):
73 7.a-7.b-re helyes válaszért 1-1, 7.c-7.e-re 2-2 pont jár.
74 Ha foldl-t ***következétesen*** foldr-nek értelmezi, 7.a-7.b-re
75 nem, de 7.c-7.e-re megkaphatja a pontokat.
76
77
78 Megjegyzések:
79 sf (ms, ns) = az ms-beli füzérek a listabeli előfordulásukhoz képest
80 fordított sorrendben egymáshoz és mind ns elé fűzve
81 f1 (ms, ns) ts = az mi^ni füzérek eredetihez képest fordított sorrendű
82 és ts elé fűzött listája, ahol mi ms, ni ns i-edik eleme,
83 és 0 <= i < min(length ms, length ns)
84 f2 (ms, ns) t = az mi^ni füzérek eredetihez képest fordított sorrendű
85 listájának összefűzésével a t elé megfordítva fűzött
86 fűzér, ahol mi ms, ni ns i-edik eleme,
87 és 0 <= i < min(length ms, length ns)
88
89 8. Tekintsük az alábbi adattípus-deklarációt:
90 datatype 'a D = L of 'a D list | P of 'a
91
92 Dombnak nevezzük az olyan (x, y, z) hármast, amelyre x < y > z. Írjon
93 olyan függvényt dombok néven, amely egy (int * int * int) D típusú
94 adatszerkezetben található dombok y magasságának az összegét adja
95 eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény
96 alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.
97
98 (* dombok : (int * int * int) D -> int
99 dombok t = a t-beli dombok magasságának összege *)
100
101 Példák:
102
103 dombok(L[]) = 0;
104 dombok(P(3,4,4)) = 0;
105 dombok(P(3,4,2)) = 4;
106 dombok(L[L[],P(2,5,8),L[],P(3,4,3)]) = 4;
107 dombok(L[L[P(3,5,4),P(6,9,9),L[L[P(8,9,8)],P(~2,3,2)]]],L[],P(3,4,5))] = 17;
108
109 Egy megoldás foldl-lel:
110
111 fun dombok (P(x,y,z)) = if x<y andalso y>z then y else 0
112 | dombok (L hs) = foldl (fn (x, y) => dombok x + y) 0 hs
113
114 Hasonló megoldás map-pel és foldl-lel:
115
116 fun dombok (P(x,y,z)) = if x<y andalso y>z then y else 0
117 | dombok (L xs) = foldl opt+ 0 (map dombok xs)
118
119 Pontozás (összesen max. 8 pont):
120 Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont
121 levonás. Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy
122 a végtelen rekurzió (-3 pont). Segédfüggvényben a fejkomment hiánya: -2
123 pont, de nem követelmény a fejkommentben a függvény típusának
124 specifikálása. Ha nagyon rossz a program hatékonysága, -2 pont a
125 levonás.
126
127

```

```

128 Deklaratív programozás nagyzárt helyi
129 Budapest, 2006. május 11.
130 =====
131 SML megoldások, V1.0, dp06s-zh2-mlmegol.txt
132 -----
133
134 B csoport
135 -----
136
137 5. Az alábbi, egymástól független, szintaktikailag helyes SML-kifejezésekben
138 kifejezésenként két-két statikus szemantikai hiba van. Melyek ezek?
139
140 (a) [(2.3, 3.2) < (3.2, 2.3)], "a" < "#b", 7 mod 3 = 0 = false
141 - string ("a") és char ("b") nem hasonlítható össze
142 - párok - (2.3,3.2) és (3.2,2.3) - kisebb-nagyobb volta nem vizsgálható
143
144 (b) (93.0, 3*4, 12, 0) = (ord "#b", 5+7, 0-7-5)
145 - négyes és hármas nem hasonlítható össze
146 - ord eredménye int, nem hasonlítható össze real-lel (93.0)
147
148 (c) List.filter [1.4, 0.6, 3.4, 4.7, 2.1, 1.9] (op /)
149 - a paraméterek sorrendje helyes: List.filter p xs
150 - p helyén predikátumnak kellene lennie, (op /) nem az
151
152 Pontozás (összesen max. 7 pont):
153 5.a - 5.c: helyes válasz 2-2-3 pont.
154 Minden hiba megtalálása 1 pontot ér, kivéve az 5.c-t, ahol a nem
155 megfelelő operátor használatának felismerésére 2 pontot adunk.
156
157 6. Mi a p értéke az alábbi, egymástól független deklarációk kiértékelése
158 után?
159
160 (a) val (_, : : p) = [{"b", "#e"}, {"l"}] @ rev(explode "tér")
161 p = [{"l"}, {"r"}, {"é"}, {"t"}]
162
163 (b) val (_, : : p :: _) = List.filter Char.isLower (explode "ld2a3r4a")
164 p = {"r"}
165
166 (c) val p = #2(foldr (fn (x,(y,b)) => (x,b andalso x>y)) (0,true) [3,2,1])
167 p = true
168
169 Pontozás (összesen max. 7 pont):
170 6.a - 6.c: helyes válasz 2-2-3 pont. Hibákért 1-1 pont levonás.
171
172 7. Tekintsük a következő függvénydefiníciókat!
173
174 (* sg : string list * string -> string *)
175 (* g1 : string list * string list -> string list -> string list *)
176 (* g2 : string list * string list -> string -> string *)
177 fun sg (ms, ns) = foldr op^ ns ms
178 and g1 (m::ms, n::ns) rs = g1 (ms, ns) (m^n::rs)
179 | g1 rs = rs
180 and g2 mens t = sg(g1 mnsns [], t)
181
182 Mi az x értéke az alábbi, egymástól független deklarációk kiértékelése
183 után?
184
185 (a) val x = sg (["n", "a", "pa"], "ma")
186 x = "napama"
187
188 (b) val x = g1 (["a", "b", "c"], [",", ","]) []
189 x = ["b", "a"]
190
191 (c) val x = g2 (["a", "b", "c"], [",", ","]) "d"
192 x = "bad"
193
194 (d) val x = g2 (["a", "a", "a"], [{"n", "r"}]) "y"
195 x = "arany"
196
197 (e) val x = let val f = map str o explode in g2 (f "oopp", f "glr") "" end
198 x = "prolog"

```

```

199
200 Pontozás (összesen max. 8 pont):
201 7.a-7.b-re helyes válaszért 1-1, 7.c-7.e-re 2-2 pont jár.
202 Ha foldr-t ***következézetesen*** foldl-nek értelmezi, 7.a-7.b-re
203 nem, de 7.c-7.e-re megkaphatja a pontokat.
204
205 Megjegyzések:
206 sg (ms, ns) = az ms-beli füzérek a listabeli előfordulásukhoz képest
207 egyenes sorrendben egymáshoz és mind ns elé fűzve
208 g1 (ms, ns) ts = az mi^ni füzérek eredetihez képest fordított sorrendű
209 és ts elé fűzött listája, ahol mi ms, ni ns i-edik eleme,
210 és 0 <= i < min(length ms, length ns)
211 g2 (ms, ns) t = az mi^ni füzérek eredetihez képest fordított sorrendű
212 listájának összefűzésével a t elé fűzött
213 fűzár, ahol mi ms, ni ns i-edik eleme,
214 és 0 <= i < min(length ms, length ns)
215
216 8. Tekintsük az alábbi adattípus-deklarációt:
217
218 datatype 'a V = S of 'a V list | P of 'a
219
220 Völgynek nevezzük az olyan (x, Y, z) hármast, amelyre x > Y < z. Írjon
221 olyan függvényt volgyek néven, amely egy (int * int * int) V típusú
222 adatszerkeztűrában található völgyek Y mélységének összegét adja
223 eredményül. Törekedjék hatékony megoldásra, magasabb rendű függvény
224 alkalmazására. Segédfüggvényt definiálhat, ha ír hozzá fejkommentet.
225
226 (* volgyek : (int * int * int) V -> int
227 volgyek t = a t-beli völgyek mélységének összege *)
228
229 Példák:
230
231 volgyek(S[]) = 0;
232 volgyek(P(3,4,4)) = 0;
233 volgyek(P(3,2,4)) = 2;
234 volgyek(S[S[],P(5,2,8)],S[],P(3,4,3))] = 2;
235 volgyek(S[S[P(5,3,4)],P(6,9,9)],S[S[P(8,9,8)],P(3,-2,2)],S[],P(4,3,5)]) = 4;
236
237 Egy megoldás foldl-lel:
238
239 fun volgyek (P(x,Y,z)) = if x>Y andalso y<z then Y else 0
240 | volgyek (S xs) = foldl (fn (x, y) => volgyek x + y) 0 xs
241
242 Hasonló megoldás map-pel és foldl-lel:
243
244 fun volgyek (P(x,Y,z)) = if x>Y andalso y<z then Y else 0
245 | volgyek (S xs) = foldl op+ 0 (map volgyek xs)
246
247 Pontozás (összesen max. 8 pont):
248 Minden kisebb hibáért 1-1 pont, minden súlyos hibáért 2 vagy 3 pont
249 levonás. Súlyos hibának számít pl. egy else ág elhagyása (-2 pont) vagy
250 a végtelen rekurzió (-3 pont). Segédfüggvényben a fejkomment hiánya: -2
251 pont, de nem követelmény a fejkommentben a függvény típusának
252 specifikálása. Ha nagyon rossz a program hatékonysága, -2 pont a
253 levonás.
254
255

```