

Deklaratív programozás, pótzárthelyi, 2003. május 6. 19.15–20.45

Munkaidő: 90 perc, összpontszám: 60

Prolog, „A” csoport (30 pont)

A Prolog-eljárás megírását kérő feladatokban a jegyzetben szereplő összes Prolog eljárás (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 2.b) hivatkozzon!

1. Döntse el, mi lesz az alábbi Prolog kérdések eredménye (hiba, meghiúsulás, siker)! Siker esetén adja meg a keletkező változó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőknek. (5 pont)

(a) $[X, Y | Z] = [a, b, c]$.

(b) $X = 2 * 3, \setminus + X = 6$.

(c) $X ::= 3 * 2, X = 6$.

(d) $X * Y = a * b + c$.

(e) $a - b - c = X - Y$.

2. Írja fel az alábbi egyenlőségek bal- és jobboldalának alapstruktúra alakját, vagy rajzolja fel a fastruktúrájukat! Adja meg, milyen változó-behelyettesítéseket eredményeznek ezek az egyesítések! (9 pont)

(a) $[4/2+X, 1] = [Y+3 | Z]$.

(b) $f([a*b+c | U], V*W) = f([K+L, L], K)$.

3. Tegyük fel, hogy az alábbi programot betöltöttük a Prolog rendszerbe.

```
p([X|_], X, X).
p([X|L], Y, Z) :-
    Y1 is Y+X, p(L, Y1, Z).
```

Állapítsa meg, hogy a feltett kérdésekre válaszul a rendszer milyen behelyettesítést ad az X változónak! Sorolja fel az összes megoldást, a rendszer által előállított sorrendben és írja le ezeket pontosvesszővel elválasztva! Ha nincs megoldás, írjon {no}-t!

(a) $p([5], 3, X)$.

(b) $p([3, 3], 0, X)$.

(c) $p([1, 2], 1, X)$.

(d) $p([1, 3, 4, 10], 2, X)$.

(e) $p([0, 2, 4, 6, 1, 13, 26], 0, X)$.

Tekintse a fenti eljárásra épülő alábbi eljárást:

```
% p(L, X): Az L listának X egy olyan eleme,....
p(L, X) :- p(L, 0, X).
```

- (f) Írja le a p/2 eljárás jelentését deklaratív módon, azaz egészítse ki teljes kijelentő mondatra a fenti fejkommentet! Írja le azt is, hogy az eljárás milyen sorrendben állítja elő a megoldásokat! (8 pont)

4. Adott két, nem feltétlenül azonos hosszú lista. Írjon olyan Prolog eljárást fesult néven, amely balról jobbra haladva összefésüli a két listát! Az eredmény-lista elemei: az első lista 1. eleme, a második lista 1. eleme, az első lista 2. eleme, a második lista 2. eleme stb. Amikor az egyik bemenő lista elemei elfogynak, a másik lista fennmaradó elemeivel zárul az eredmény. Vigyázzon arra, hogy az eljárás ne adjon hamis illetve többszörös eredményt! Segédeljárást nem definiálhat. (8 pont)

% fesult(+L1, +L2, -F): Az F lista az L1 és L2 listák összefésültje.

```
| ?- fesult([], [], L).           => L = [] ? ; no
| ?- fesult([1,2], [3,4,5,6,7], L). => L = [1,3,2,4,5,6,7] ? ; no
| ?- fesult([a,b,c,d], [e,f,g], L). => L = [a,e,b,f,c,g,d] ? ; no
```

Deklaratív programozás, pótzárthelyi, 2003. május 6. 19.15–20.45

Munkaidő: 90 perc, összpontszám: 60

Standard ML, „A” csoport (30 pont)

Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa (az aritmetikai függvények és a relációk kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
map	: ('a -> 'b) -> 'a list -> 'b list	implode	: char list -> string
op::	: 'a * 'a list -> 'a list	String.sub	: string * int -> char
chr	: int -> char	hd	: 'a list -> 'a
ord	: char -> int	tl	: 'a list -> 'a list

5. Mi az **r** típusa az alábbi, egymástól független deklarációk kiértékelése után? (6 pont)

- (a) `fun r (x, y) = (y, y x)`
- (b) `fun r x y = ((x y) y)`
- (c) `fun r x y = (x, x y)`

6. Mi a **v** értéke és típusa az alábbi, egymástól független deklarációk kiértékelése után? (6 pont)

- (a) `val (v::w::_) = tl(explode "irat")`
- (b) `val (v, "v") = hd(List.filter (fn (x,y) => y="v")
[("va", "vk"), ("v", "z"), ("5", "v")])`
- (c) `val v = implode(map (fn x => chr(ord x + 2)) [# "A", # "F", # "R"])`

7. Az 'a drink adattípust így deklaráljuk: `datatype 'a drink = Water | Cola of 'a`

(a) Írja fel `Water`, `Cola` és `Cola "Light"` típusát!

Tekintsük az `f` függvény definícióját:

```
fun f Water = 0
  | f (Cola "Light") = 199
  | f (Cola s) = if String.sub(s,0) < # "M" then 149 else 249
```

(b) Mi az **x** értéke az alábbi, egymástól független deklarációk kiértékelése után?

- (b1) `val x = f Water`
- (b2) `val x = f (Cola "Fanta")`
- (b3) `val x = f (Cola "Light")`
- (b4) `val x = f (Cola "Super")`

(c) Specifikálja az `f` függvény típusát és értelmezési tartományát! (9 pont)

8. Írjon SML-függvényt `torlodok` néven, amely megszámlolja, hogy egy magyar szóban hány helyen van mássalhangzótorlódás. Mássalhangzótorlódásnak nevezzük a három szomszédos mássalhangzóból álló betűcsoportot, amelyben az egymás mellett állók betűk nem azonosak. A többjegyű mássalhangzók (pl. a cs és a gy) minden betűjét külön mássalhangzónak tekintjük. A vizsgálandó szavakban csak betűk fordulnak elő. Nem kell hatékony megoldásra törekednie. Segédfüggvényt definiálhat (fejkommenttel!).

Használhatja a `massal = "bBcCdDfFgGhHjJkKlLmMnNpPqQrRsStTvVwWxXyYzZ"` értéket, továbbá az `isMem : 'a * 'a list -> bool` tesztelő függvényt annak vizsgálatára, hogy egy adott érték benne van-e egy adott listában. (9 pont)

```
(* torlodok : string -> int
   torlodok s = az s szóban lévő hármas mássalhangzótorlódások száma
*)
```

Példák: `torlodok "vegyszerboltban" = 3;` `torlodok "kettoshangzo" = 1;`
`torlodok "gyümölcstorta" = 2;` `torlodok "pattogatott" = 0;`
`torlodok "gyümölcsszörppel" = 1;` `torlodok "" = 0;`

Deklaratív programozás, pótzárthelyi, 2003. május 6. 19.15–20.45

Munkaidő: 90 perc, összpontszám: 60

Prolog, „B” csoport (30 pont)

A Prolog-eljárás megírását kérő feladatokban a jegyzetben szereplő összes Prolog eljárás (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 2.b) hivatkozzon!

1. Döntse el, mi lesz az alábbi Prolog kérdések eredménye (hiba, meghiúsulás, siker)! Siker esetén adja meg a keletkező változó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőknek. (5 pont)

(a) $X = 2+3, 5 = X.$

(b) $Y \text{ is } X+1, X \text{ is } 3*4.$

(c) $[X|[Y]] = [a,b].$

(d) $U \text{ is } 2+4, V = U+2.$

(e) $P+Q = 48/12+3.$

2. Írja fel az alábbi egyenlőségek bal- és jobboldalának alapstruktúra alakját, vagy rajzolja fel a fastruktúrájukat! Adja meg, milyen változó-behelyettesítéseket eredményeznek ezek az egyesítések! (9 pont)

(a) $h(X, [X,1]) = h(B+A, [8*2+1|C]).$

(b) $[f(P,a*b), Q|R] = [Q, f(6,S), S].$

3. Tegyük fel, hogy az alábbi programot betöltöttük a Prolog rendszerbe.

```
r([X|L], _, Z) :-  
    r(L, X, Z).  
r([X,Y|_], Z, X) :- 2*X == Y+Z.
```

Állapítsa meg, hogy a feltett kérdésekre válaszul a rendszer milyen behelyettesítést ad az X változónak! Sorolja fel az összes megoldást, a rendszer által előállított sorrendben és írja le ezeket pontosvesszővel elválasztva! Ha nincs megoldás, írjon {no}-t!

(a) $r([2], 2, X).$

(b) $r([2,3], 1, X).$

(c) $r([2,2,2], 2, X).$

(d) $r([3,5,4,3], 1, X).$

(e) $r([3,4,2,0,5,10,15,2], 2, X).$

Tekintse a fenti eljárásra épülő alábbi eljárást:

```
% r(L, Y): Az L listának Y egy olyan eleme,....  
r([X|L], Y) :-  
    r(L, X, Y).
```

- (f) Írja le az r/2 eljárás jelentését deklaratív módon, azaz egészítse ki teljes kijelentő mondatná a fenti fejkommentet! Írja le azt is, hogy az eljárás milyen sorrendben állítja elő a megoldásokat! (8 pont)

4. Adott két, nem feltétlenül azonos hosszú számlista. Írjon olyan Prolog eljárást *osszeg* néven, amely egy olyan listát állít elő, amely a két adott listának, mint vektornak az összege! Ha az adott listák nem azonos hosszúságúak, akkor a rövidebbik listát megfelelő számú 0-val kell a végén kiegészíteni. Tehát az eredmény-lista hossza a két adott lista hosszának maximuma. Vigyázzon arra, hogy az eljárás ne adjon hamis illetve többszörös eredményt! Segédeljárást nem definiálhat. (8 pont)

```
% osszeg(+L1, +L2, -S): Az S lista az L1 és L2 listák összege.
```

```
| ?- osszeg([], [], L).           => L = [] ? ; no  
| ?- osszeg([1,2], [3,4,5,6,7], L). => L = [4,6,5,6,7] ? ; no  
| ?- osszeg([1,2,3,4], [5,6,7], L). => L = [6,8,10,4] ? ; no
```

Deklaratív programozás, pótzárthelyi, 2003. május 6. 19.15–20.45
Munkaidő: 90 perc, összpontszám: 60
Standard ML, „B” csoport (30 pont)

Az SML-függvény megírását kérő feladatokban a jegyzetben szereplő összes SML-függvény (akár beépített, akár a jegyzetben definiált) szabadon használható. Az egyes részfeladatokra szám- és betűjelükkel (pl. 6.b) hivatkozzon!

A feladatokban előforduló könyvtári függvények típusa (az aritmetikai függvények és a relációk kivételével):

List.filter	: ('a -> bool) -> 'a list -> 'a list	explode	: string -> char list
map	: ('a -> 'b) -> 'a list -> 'b list	implode	: char list -> string
op:::	: 'a * 'a list -> 'a list	String.sub	: string * int -> char
chr	: int -> char	hd	: 'a list -> 'a
ord	: char -> int		

5. Mi a **t** típusa az alábbi, egymástól független deklarációk kiértékelése után? (6 pont)

- (a) fun t (x, y) = (x, (x, y))
- (b) fun t x y = (x y, x y y)
- (c) fun t y x = ((y x) x)

6. Mi a **v** értéke és típusa az alábbi, egymástól független deklarációk kiértékelése után? (6 pont)

- (a) val ((_:::_::v)::_) = [explode "irat"]
- (b) val ("v", v) = hd(List.filter (fn (x,y) => x="v")
[("va", "vk"), ("v", "z"), ("5", "v")])
- (c) val w = implode(map (fn x => chr(ord x - 2)) [#"C", #"H", #"S"])

7. Az 'a food adattípust így deklaráljuk: datatype 'a food = Air | Meat of 'a

(a) Írja fel Air, Meat és Meat "Frog" típusát!

Tekintsük az f függvény definícióját:

```
fun f Air = 0
  | f (Meat "Frog") = 219
  | f (Meat t) = if String.sub(t,0) > #"M" then 249 else 189
```

(b) Mi az **x** értéke az alábbi, egymástól független deklarációk kiértékelése után?

- (b1) val x = f Air
- (b2) val x = f (Meat "Chicken")
- (b3) val x = f (Meat "Frog")
- (b4) val x = f (Meat "Rind")

(c) Specifikálja az f függvény típusát és értelmezési tartományát! (9 pont)

8. Írjon SML-függvényt hangrend néven, amely meghatározza, hogy egy magyar szó magas, mély vagy vegyes hangrendű-e. A függvény eredménye legyen 1, ha az argumentuma magas, ~1, ha mély, és 0, ha vegyes hangrendű szó. Egy magas hangrendű szóban vannak magas hangrendű magánhangzók (e, é, ö, ő, ü, ű), de nincsenek mély hangrendűek (a, á, o, ó, u, ú). Egy mély hangrendű szóban vannak mély hangrendű magánhangzók, de nincsenek magas hangrendűek. Minden más szó (az üres szó is!) vegyes hangrendű. Az i és í magánhangzók a hangrend szempontjából közömbösek. Nem kell hatékony megoldásra törekednie. Segédfüggvényt definiálhat (fejkommenttel!).

Használhatja a magas="EeÉÉöÖőŐüÜűŰ" és mely="aAáÁoOóÓuUúÚ" értékeket, továbbá az isMem : 'a * 'a list -> bool tesztelő függvényt annak vizsgálatára, hogy egy adott érték benne van-e egy adott listában. (9 pont)

```
(* hangrend : string -> int
   hangrend s = 1, ha magas, ~1, ha mély, 0, ha vegyes hangrendű szó az s
*)
```

Példák: hangrend "örömszülő" = 1; hangrend "méregpohár" = 0;
 hangrend "kitekintő" = 1; hangrend "licit" = 0;
 hangrend "ormótlan" = ~1; hangrend "psszt" = 0;
 hangrend "viharos" = ~1; hangrend "" = 0;