

Deklaratív programozás, nagyvárthelyi, 2003. március 24. 16.15–17.45

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, megghiúsulás, siker)! Siker esetén adja meg a keletkező változó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőknek. (5 pont)

(a) Y is $2*3$, $Z = Y+1$.

(b) $[A,B] = [x,y,z]$.

(c) $X+Y = 2+3*4$.

(d) $X = 2*4$, $\backslash+$ $X = 8$.

(e) $X ::= 3*4$.

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) $f([U|V], 1+2+3) = f([a,b], X+Y)$

(b) $.(f(A+C,B), [A|B]) = [f(2*3+4, [p,q]) | Z]$.

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

```
p([X|_], Y, X) :- X > Y.
```

```
p([_,X|L], _, Z) :-
```

```
    p(L, X, 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 pontos vesszővel elválasztva! Ha nincs megoldás, írjon {no}-t!

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

(b) $p([3], 1, X)$.

(c) $p([1,2,4], 5, X)$.

(d) $p([3,1,2,7], 1, X)$.

(e) $p([2,5,3,4,7,3,5,8], 0, X)$.

Tekintse a fenti eljárásra épülő alábbi eljárást:

```
% p(L, Y): Az L listának Y egy olyan eleme,....
```

```
p([X|L], Y) :- p(L, X, Y).
```

- (f) Írja le a $p/2$ 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. Egy számlistában *csúcs*nak hívunk három szomszédos elemet, ha az első és a harmadik kisebb a másodiknál. Írjon olyan Prolog eljárást *csucs*a néven, amely balról jobbra haladva felsorolja egy számlistában levő csúcsokat! A három szomszédos elemet egy *cs/3* funktorú struktúraként adja vissza! Segédeljárást nem definiálhat. (8 pont)

```
% csucs(+L, -Cs): Cs egy cs/3 funktorú struktúra, amely az L számlistában
```

```
% levő csúcst ír le. L bemenő, Cs kimenő paraméter.
```

```
| ?- csucs([1,2,3], Cs).           => no
```

```
| ?- csucs([1,2,3,1,4,2], Cs).    => Cs = cs(2,3,1) ? ; Cs = cs(1,4,2) ? ; no
```

```
| ?- csucs([1,2,3,-3,2,1,0,2,7,5,8,9,1], Cs).
```

```
                                => Cs = cs(2,3,-3) ? ; Cs = cs(-3,2,1) ? ;
```

```
                                Cs = cs(2,7,5) ? ; Cs = cs(8,9,1) ? ; no
```

Pluszfeladat: Írjon az alábbi fejkommentnek megfelelő Prolog eljárást! Segédeljárást definiálhat, de fejkommentet feltétlenül adjon meg hozzá. (+3 pont)

```
% csucs(+L, -I, -Cs): Cs az L számlista I.-edik csúcsa. L bemenő, I és Cs kimenő.
```

```
| ?- csucs([1,2,3,-3,2,1,0,2,7,5,8,9,1], I, Cs).
```

```
                                => I = 1, Cs = cs(2,3,-3) ? ; I = 2, Cs = cs(-3,2,1) ? ; ...
```

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	str	: char -> string
chr	: int -> char	hd	: 'a list -> 'a
ord	: char -> int		

5. Mi a **h típusa** az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

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

6. Mi a **w értéke** és **típusa** az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

- (a) val (_::w::_) = [# "w", # "v", # "W", # "9", # ":"]
- (b) val ("al", w) = hd(List.filter (fn (x, _) => x="al") [("lo", 2), ("al", ~3), ("al", 1)])
- (c) val w = implode(map (fn x => chr(ord x + 1)) (explode "abcd"))

7. Az 'a lap adattípust így deklaráljuk: datatype 'a lap = Joker | Piros of 'a

- (a) Írja fel Joker, Piros és Piros "Ász" típusát!

Tekintsük az f függvény definícióját:

```
fun f Joker = ""
  | f (Piros 7) = "99"
  | f (Piros n) = str(chr n)
```

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

- (b1) val x = f Joker
- (b2) val x = f (Piros 7)
- (b3) val x = f (Piros(ord # "A"))

- (c) Írjon fejkommentet az f függvényhez, azaz specifikálja a típusát és az értelmezési tartományát, továbbá írja le deklaratív módon az eredményét! (7 pont)

8. Írjon olyan SML-függvényt csucsock néven, amely egy egész számokat tartalmazó listából (balról jobbra haladva) kigyűjti az összes olyan, egymással szomszédos három számot számhármasként, amelyekben az első és a harmadik szám kisebb a másodiknál. Segédfüggvényt definiálhat (fejkommenttel!). (9 pont)

```
(* csucsock : int list -> (int * int * int) list
   csucsock xs = az összes olyan xs-beli, egymással szomszédos három számból
                 álló hármasként listája (balról jobbra haladva xs-ben),
                 amelyekben az első és harmadik szám kisebb a másodiknál
*)
```

Példák: csucsock [] = [];
csucsock [4,6] = [];
csucsock [4,9,6] = [(4,9,6)];
csucsock [4,6,9] = [];
csucsock [4,9,~4,0,~2,4,9,2,3,6] = [(4,9,~4), (~4,0,~2), (4,9,2)];

Deklaratív programozás, nagyzárthelyi, 2003. március 24. 16.15–17.45

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, megghiúsulás, siker)! Siker esetén adja meg a keletkező változó-behelyettesítéseket! A kérdéseket egyenként és önmagukban adjuk át az értelmezőknek. (5 pont)

- (a) $Z \text{ is } Y+1, Y \text{ is } 2*3.$
- (b) $[U|V] = [a,b].$
- (c) $X*Y = 2+3*4.$
- (d) $X \text{ is } 1+2, X+9 ::= 3*4.$
- (e) $F*G = a*b*c.$

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) $[U, 1*2*3|V] = [a,X*Y,c].$
- (b) $f(W+a*b,[2|Z]) = f(P+Q,[P|(Q,[s])]).$

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

```
r([X|_], Y, X) :- X =< Y.  
r([X,_|L], _, Z) :-  
    r(L, X, 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) $r([], 3, X).$
- (b) $r([3], 4, X).$
- (c) $r([2,3,4], 5, X).$
- (d) $r([3,1,2,7], 7, X).$
- (e) $r([5,2,8,4,7,3,4,2], 9, X).$

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. Egy számlistában *kút*nak hívunk három szomszédos elemet, ha az első és a harmadik ugyanannyival nagyobb a második elemnél.

Írjon olyan Prolog eljárást *kutja* néven, amely balról jobbra haladva felsorolja egy számlistában levő kútakat! A három szomszédos elemet egy $k/3$ funktorú struktúraként adja vissza! Segédeljárást nem definiálhat. (8 pont)

```
% kutja(+L, -K): K egy k/3 funktorú struktúra, amely az L számlistában  
% levő kútat ír le. L bemenő, K kimenő paraméter.
```

```
| ?- kutja([1,2,3], K).           => no  
| ?- kutja([3,1,3,2], K).       => K = k(3,1,3) ? ; no  
| ?- kutja([1,2,3,-3,3,1,0,1,-5,1,9], K).  
    => K = k(3,-3,3) ? ; K = k(1,0,1) ? ; K = k(1,-5,1) ? ; no
```

Pluszfeladat: Írjon az alábbi fejkommentnek megfelelő Prolog eljárást! Segédeljárást definiálhat, de fejkommentet feltétlenül adjon meg hozzá. (+3 pont)

```
% kutja(+L, -I, -K): K az L számlista I.-edik kútja. L bemenő, I és K kimenő.
```

```
| ?- kutja([1,2,3,-3,3,1,0,1,-5,1,9], I, K).  
    => I = 1, K = k(3,-3,3) ? ; I = 2, K = k(1,0,1) ? ; ...
```

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	str	: char -> string
chr	: int -> char	hd	: 'a list -> 'a
ord	: char -> int		

5. Mi a **k** típusa az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

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

6. Mi a **w** értéke és típusa az alábbi, egymástól független deklarációk kiértékelése után? (7 pont)

- (a) val [_,_,(w:_)] = [[#"W"], [#"w", #"Z"], [#"v", #"z"]]
- (b) val (w,"ma") = hd(List.filter (fn (_,y) => y="ma") [("jo","fa"),("al","ma"),("el","ma")])
- (c) val w = implode(map (fn x => chr(ord x - 1)) (explode "WXYZ"))

7. Az 'a lap adattípust így deklaráljuk: datatype 'a lap = Pagat | Makk of 'a

- (a) Írja fel Pagat, Makk és Makk "Filkó" típusát!

Tekintsük az f függvény definícióját:

```
fun f Pagat = "1"
  | f (Makk 7) = "Németvölgyi út 49."
  | f (Makk n) = str(chr n)
```

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

- (b1) val x = f Pagat
- (b2) val x = f (Makk 7)
- (b3) val x = f (Makk(ord #"A"))

- (c) Írjon fejkommentet az f függvényhez, azaz specifikálja a típusát és az értelmezési tartományát, továbbá írja le deklaratív módon az eredményét! (7 pont)

8. Írjon olyan SML-függvényt **kutak** néven, amely egy egész számokat tartalmazó listából (balról jobbra haladva) kigyűjti az összes olyan, egymással szomszédos három számot számhármassok listájaként, amelyekben az első és a harmadik szám ugyanannyival nagyobb a másodiknál. Segédfüggvényt definiálhat (fejkommenttel!). (9 pont)

```
(* kutak : int list -> (int * int * int) list
   kutak xs = az összes olyan xs-beli, egymással szomszédos három számból álló
              hármassoknak a listája, (balról jobbra haladva xs-ben), amelyekben
              az első és harmadik szám ugyanannyival nagyobb a másodiknál
*)
```

Példák: kutak [] = [];
 kutak [6,4] = [];
 kutak [6,4,6] = [(6,4,6)];
 kutak [6,4,9] = [];
 kutak [4,2,4,0,4,4,9,~2,9,6] = [(4,2,4), (4,0,4), (9,~2,9)];