

Programare logică și funcțională

- examen scris -

Notă

1. Subiectele se notează astfel: of - 1p; A - 1.5p; B - 2.5p; C - 2.5p; D - 2.5p.
2. Problemele Prolog vor fi rezolvate în SWI Prolog. Se cere: (1) explicarea codului și a raționamentului; (2) modelul recursiv de rezolvare, pentru fiecare predicat folosit; (3) specificarea fiecărui predicat (semnificația parametrilor, model de flux, tipul predicatului - determinist/nedeterminist).
3. Problemele Lisp vor fi rezolvate în Common Lisp. Se cere: (1) explicarea codului și a raționamentului; (2) modelul recursiv de rezolvare, pentru fiecare funcție folosită; (3) specificarea fiecărei funcții (semnificația parametrilor).

A. Fie următoarea definiție de predicat PROLOG **f(integer, integer)**, având modelul de flux (i, o):

$f(0, 0):-!$.

$f(I,Y):-J \text{ is } I-1, \text{ f(J,V), } V>1, !, K \text{ is } I-2, Y \text{ is } K.$

$f(I,Y):-J \text{ is } I-1, \text{ f(J,V), } Y \text{ is } V+1.$

Rescrieți această definiție pentru a evita apelul recursiv **f(J,V)** în ambele clauze. Nu redefiniți predicatul. Justificați răspunsul.

- B.** Dându-se o listă neliniară care conține atomi numerici și nenumeriți, se cere un program Lisp care verifică dacă secvența atomilor numerici de pe toate nivelurile impare formează o secvență zig-zag (elementul 2 e mai mare decât primul element, elementul 3 este mai mic decât elementul 2, elementul 4 este mai mare decât elementul 3, etc.). De exemplu, pentru lista (10 21 (3 A (B (0 77) 1 77)) C (5 (D 54) 11 6) 89 F H) rezultatul va fi true (secvența zig-zag este (10 21 1 77 54 89)).

- C. Să se scrie un program PROLOG care generează lista submulțimilor formate cu elemente unei liste listă de numere întregi, având număr suma elementelor număr impar și număr par nenul de elemente pare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista $[2,3,4] \Rightarrow [[2,3,4]]$

- D. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2). Se cere să se determine lista nodurilor de pe nivelurile pare din arbore (în ordinea nivelurilor 0, 2, ...). Nivelul rădăcinii se consideră 0. **Se va folosi o funcție MAP.**
- Exemplu** pentru arborele (a (b (g)) (c (d (e)) (f))) => (a g d f)