

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(list, integer)**, având modelul de flux (i, o):

$f([], -1) :- !.$

$f([_|T], Rez) :- \text{f(T,S)}, S < 1, !, Y \text{ is } S+2.$

$f([H|T], Rez) :- \text{f(T,S)}, S < 0, !, Y \text{ is } S+H.$

$f([_|T], Rez) :- \text{f(T,S)}, Y \text{ is } S.$

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

- B.** Dându-se o listă neliniară care conține atomi numerici și nenumeriți, se cere un program Lisp, care construiește o listă care are aceeași structură ca lista inițială, dar fiecare (sub)listă conține un singur element pe nivel superficial, și acel element este diferența dintre valoarea maximă și cea minimă de pe nivelul superficial al (sub)listei corespunzătoare în lista inițială. Este garantat că fiecare (sub)listă conține măcar un atom numeric. De exemplu, pentru lista (F A 12 13 (B 11 (A D 15) C 3 C (1 F 6) 1) 18 11 D (A 7 F 9) F)) rezultatul va fi (7 (10 (0) (5)) (2)) (7 este diferența dintre 18 și 11, 10 este diferența dintre 11 și 1, 0 este diferența dintre 15 și 15, etc.).

- C. Să se scrie un program PROLOG care generează lista aranjamentelor de **k** elemente dintr-o listă de numere întregi, având o sumă **S** dată. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista [6, 5, 3, 4], **k**=2 și **S**=9 \Rightarrow [[6,3],[3,6],[5,4],[4,5]] (nu neapărat în această ordine)

- D. Se consideră o listă neliniară. Să se scrie o funcție LISP care să aibă ca rezultat lista inițială din care au fost eliminați toți atomii de pe nivelul **k** (nivelul superficial se consideră 1). **Se va folosi o funcție MAP.**

Exemplu pentru lista (a (1 (2 b)) (c (d)))

a) k=2 => (a ((2 b)) ((d))) **b)** k=1 => ((1 (2 b)) (c (d))) **c)** k=4 => lista nu se modifică