

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 L o listă numerică și următoarea definiție de predicat PROLOG având modelul de flux (i, o):

$f([], -1)$.

$f([H|T], S) :- f(T, S1), S1 > 0, !, S \text{ is } S1 + H$.

$f([_|T], S) :- f(T, S1), S \text{ is } S1$.

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

- B.** Dându-se o listă neliniară conținând atât atomi numerici cât și nenumeriți, să se scrie un program LISP care calculează cel mai mare divizor comun al acelor numere care se află între 2 atomi nenumeriți (vecinii unui număr se consideră la orice nivel). **De exemplu**, pentru lista (A B 12 (5 D (A F (15 B) D (5 F) 4)) C 9) numerele al căror cel mai mare divizor comun trebuie calculat sunt: 15 (între F și B), 5 (între D și F) și 4 (între F și C), iar rezultatul va fi 60. Nu este permisă utilizarea funcției predefinite *gcd* din Lisp.

- C. Scrieți un program PROLOG care determină dintr-o listă formată din numere întregi lista subșirurilor cu cel puțin 2 elemente, formate din elemente în ordine strict crescătoare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista [1, 8, 6, 4] \Rightarrow [[1,8],[1,6],[1,4],[6,8],[4,8],[4,6],[1,4,6],[1,4,8],[1,6,8],[4,6,8],[1,4,6,8]] (nu neapărat în această ordine)

- D. Se consideră o listă neliniară. Să se scrie o funcție care să aibă ca rezultat lista inițială în care atomii de pe nivelul **k** au fost înlocuiți cu 0 (nivelul superficial se consideră 1). **Se va folosi o funcție MAP.**

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

a) k=2 => (a (0 (2 b)) (0 (d))) **b)** k=1 => (0 (1 (2 b)) (c (d))) **c)** k=4 => lista nu se modifică