

# Programare Logică – MODALITATEA DE DESFĂȘURARE A EXAMENULUI ȘI EXEMPLU DE SUBIECT DE EXAMEN

Claudia MUREȘAN

UNIVERSITATEA DIN BUCUREȘTI, FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
c.muresan@yahoo.com, cmuresan@fmi.unibuc.ro

2019–2020, Semestrul I

**Modalitatea de desfășurare a examenului:** EXAMEN SCRIS, CU LISTE DE SUBIECTE INDIVIDUALE PENTRU FIECARE STUDENT TRIMISE PE MAILURILE INSTITUȚIONALE PE CARE AM TRIMIS MATERIALELE DE CURS, SEMINAR ȘI LABORATOR, ȘI PRIMIREA LUCRĂRILOR DIN PARTEA STUDENȚILOR CA FIȘIERE ATAȘATE LA UN CÂTE UN MAIL TRIMIS PE CELE DOUĂ ADRESE DE EMAIL DE MAI SUS (DIN TITLUL ACESTUI DOCUMENT).

Din momentul primirii **listei personale de subiecte** pe adresa de email instituțională pe care ați primit de la mine materialele de curs, seminar și laborator, veți avea la dispoziție **3 ore** pentru a-mi trimite rezolvările subiectelor de examen pe cele două adrese de email de mai sus, în FORMATELE INDICATE MAI JOS.

Dacă, în cel mult **30–40 de minute** de la ora începerii examenului, nu primiți de la mine, pe mail, lista de subiecte de examen, trebuie neapărat să-mi scrieți, pe aceste două adrese de email de mai sus, să-mi cereți lista de subiecte; nu contactați un coleg pentru lista de subiecte, pentru că lista sa de subiecte nu va coincide cu cea destinată vouă (doar gradul de dificultate și cantitatea de muncă necesară pentru redactarea rezolvărilor subiectelor vor fi similare).

Dacă nu ați primit de la mine materiale de curs, seminar și laborator pe **mailul personal** înregistrat pe pagina moodle a cursului (nu prin intermediul șefului de grupă sau al grupului Google al seriei ID), trebuie neapărat să-mi scrieți să vă înregistrez mailul personal în lista de mailuri pentru examen și să vă dau acces la folderul din Dropbox cu o serie de lecții de pe Zoom.

**Regulament privind formatul în care studenții trebuie să-mi trimită pe mail lucrările de examen:**

- fiecare student va include numele său și inițialele prenumelui în denumirile tuturor documentelor conținând rezolvările sale pentru subiectele de examen;
- rezolvările pentru subiectele de seminar (de tipul Exercițiilor 1 și 2 din lista de mai jos) vor fi atașate la mailul cu lucrarea de examen scanate sau fotocopyate lizibil, în format PDF/PNG/BMP/JPG;
- rezolvările pentru subiectele de programare în Prolog (precum Exercițiile 3 și 4 din următoarea listă) vor fi atașate la mailul cu lucrarea de examen atât în fișiere plain text, cât și extrase în fișiere PDF din aceste fișiere text sau decupate de pe ecran, în fișiere PNG/BMP/JPG.

Vedeți **împărțirea punctajului** la sfârșitul acestui document.

## Exemplu de listă de subiecte de examen:

**Exercițiul 1.** Considerăm un limbaj de ordinul I conținând un simbol de relație binară  $p$ , două simboluri de relații unare distincte  $q$  și  $r$ , un simbol de operație ternară  $f$ , unul de operație binară  $g$ , unul de operație unară  $h$  și un simbol de constantă  $c$ . Fie  $X$ ,  $Y$  și  $Z$  variabile distincte.

- (1) Aplicând algoritmul de unificare, să se determine dacă următorii termeni au unificator și, în caz afirmativ, să se determine un cel mai general unificator pentru aceștia:

$$f(h(h(g(c, X))), c, g(h(X), Z)) \text{ și } f(h(Z), X, Y).$$

- (2) Să se determine o formă Skolem pentru următorul enunț:

$$\forall X [\exists Y p(X, h(Y)) \vee \exists Z [q(f(c, X, h(Z))) \wedge [r(h(Z)) \rightarrow \neg q(f(c, c, h(Z)))]]]$$

și să se aplice algoritmul Davis–Putnam acelei forme Skolem.

**Exercițiul 2.** Având următoarea bază de cunoștințe în Prolog:

*fructe(mere). fructe(pere). fructe(capsuni).*  
*sunt(mere, rosii). sunt(capsuni, rosii). sunt(pere, gustoase). sunt(capsuni, gustoase).*  
*consuma(ana, X) :- fructe(X), sunt(X, rosii).*  
*consuma(victor, X) :- fructe(X), sunt(X, rosii).*  
*consuma(victor, X) :- fructe(X), sunt(X, gustoase).*

să se scrie arborele de derivare prin rezoluție SLD pentru următoarea interogare:

?- *consuma(Cine, Ce).*

și să se determine soluțiile acestei interogări folosind acest arbore de derivare.

**Exercițiul 3.** Să se scrie un predicat ternar *selectpoz(Lista, TotACataPozitie, ListaElementeSelectate)*, definit ca mai jos, precum și toate predicatele auxiliare necesare pentru definirea acestuia:

*selectpoz* să fie satisfăcut ddacă primul său argument este o listă  $L$ , al doilea argument al său este un număr natural nenul  $N$ , iar al treilea argument este lista elementelor lui  $L$  aflate în lista  $L$  pe poziții date de multiplii lui  $N$  cuprinși între  $N$  și lungimea listei  $L$ ;

și, într-o interogare în Prolog, *selectpoz* să funcționeze sub forma: dacă primește o listă arbitrară  $L$  în primul argument și un număr natural nenul arbitrar  $N$  în al doilea argument, să obțină în al treilea argument lista elementelor lui  $L$  selectate din  $N$  în  $N$  poziții, ca mai sus; de exemplu:

la interogările următoare:	Prologul să răspundă:
?- <i>selectpoz([], 1, Lista).</i>	<i>Lista = [];</i>
?- <i>selectpoz([1, 2, 3], 1, Lista).</i>	<i>Lista = [1, 2, 3];</i>
?- <i>selectpoz([1, 2, 3], 5, Lista).</i>	<i>Lista = [];</i>
?- <i>selectpoz([1, 2, 3], 2, Lista).</i>	<i>Lista = [2];</i>
?- <i>selectpoz([1, 2, 3, 4, 5, 6, 7], 3, Lista).</i>	<i>Lista = [3, 6];</i>
?- <i>selectpoz([X, a, b, 10, c, Y, 7, [1, 2, 3], V, h(X)], 2, Lista).</i>	<i>Lista = [a, 10, Y, [1, 2, 3], h(X)].</i>

**Exercițiul 4.** Să se scrie în Prolog un predicat ternar

*maxar(Termen, AritateMaxima, ListaNumeOperatiideAritateMaxima)*

definit ca mai jos, precum și toate predicatele auxiliare necesare pentru definirea acestuia:

*maxar* să fie satisfăcut ddacă primul său argument este un termen Prolog, al doilea argument al său este numărul natural egal cu maximum arităților operațiilor din componența termenului de pe primul argument, iar al treilea argument al său este:

- lista numerelor operațiilor de aritate maximă din componența acelui termen, dacă termenul nu este o variabilă;
- lista vidă, dacă acel termen este o variabilă, așadar nu are operații în componența sa, caz în care al doilea argument al lui *maxar* trebuie să aibă valoarea 0;

și, într-o interogare în Prolog, *maxar* să funcționeze sub forma: dacă primește un termen arbitrar în primul argument, să calculeze, în al doilea argument, maximul arităților operațiilor din componența acelui termen, iar, în al treilea argument, să obțină lista operațiilor de aritate maximă din componența acelui termen; de exemplu:

la interogările următoare:	Prologul să răspundă:
?- <i>maxar</i> ( <i>c</i> , <i>Aritate</i> , <i>Lista</i> ).	<i>Aritate</i> = 0, <i>Lista</i> = [ <i>c</i> ];
?- <i>maxar</i> ([], <i>Aritate</i> , <i>Lista</i> ).	<i>Aritate</i> = 0, <i>Lista</i> = [[]];
?- <i>maxar</i> ( <i>X</i> , <i>Aritate</i> , <i>Lista</i> ).	<i>Aritate</i> = 0, <i>Lista</i> = [];
?- <i>maxar</i> ([1, <i>g</i> ( <i>X</i> , <i>k</i> ( <i>X</i> , <i>Y</i> )), 3], <i>Aritate</i> , <i>Lista</i> ).	<i>Aritate</i> = 2, <i>Lista</i> = [[[]], <i>g</i> , <i>k</i> ];

iar, la interogarea:

?- *maxar*(*f*([1, 2, 3], *f*(*a*, *g*(*X*, *g*(*a*, *g*(*X*, *Y*)), *b*, *h*(*Z*), *V*)), *b*), *h*(*g*(*a*, *fct*(*h*(1), *g*(*V*, *h*(*a*))), [*a*, *b*, *c*, *h*(*h*(*g*(*A*, *B*)))]), *f*(*a*, *b*, *c*), [*fct*(1, 2, 3, 4, 5), *f*(1, 2, 3), *g*(1, 2), *g*(1, 2, 3, 4, 5)))]), *Aritate*, *Lista*).

Prologul să răspundă: *Aritate* = 5, *Lista* = [*g*, *fct*]; observăm că, în componența termenului din această interogare, avem o operație *g* de aritate 2, dar și una de aritate 5 cu aceeași denumire.

**Notă:** Dacă doriți consultații înainte de examen, să-mi scrieți pe mail și să-mi cereți o întâlnire pe Zoom, sau câte o întâlnire pe Zoom pentru fiecare serie; vă stau la dispoziție, în ce zi doriți, la orice oră după ora 16:00, dacă îmi scrieți, pe ambele mailuri, din timp: cu cel puțin două zile înainte de ziua examenului, pentru consultații în ziua care precedă ziua examenului – preferabil prima zi de examen la PL, în cazul seriei 24, dar, pentru întrebări punctuale (nu lecții recapitulative la care nu aveți nimic de întrebat), stau la dispoziție fiecărei grupe în parte.

**Punctaj:** 2 puncte din oficiu;

toate exercițiile au același punctaj: pentru fiecare dintre Exercițiile 1, 2, 3 și 4, punctajul este: (10–punctajul pentru temele colective)/4;

în cazul unui exercițiu cu mai multe subpuncte, precum Exercițiul 1 de mai sus, punctajul acelui exercițiu este împărțit în mod egal între acele subpuncte.

**Nota** va fi minimul dintre 10 și suma între punctajul din oficiu, punctajul pentru temele colective de programare în Prolog și punctajele acumulate din rezolvările tuturor exercițiilor din lista de subiecte.