

Programare Logică – LISTĂ SUBIECTE DE EXAMEN

Claudia MUREȘAN, c.muresan@yahoo.com, cmuresan@fmi.unibuc.ro

UNIVERSITATEA DIN BUCUREȘTI, FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

2019–2020, Semestrul I

Exercițiul 1. Considerăm un limbaj de ordinul I conținând un simbol de operație binară f , un simbol de operație unară g și un simbol de constantă c . Fie X , Y și Z variabile distincte.

Să se deseneze arborii de expresii asociați următorilor doi termeni, apoi, prin aplicarea algoritmului de unificare, să se determine dacă acești termeni au unificator și, în caz afirmativ, să se determine un cel mai general unificator pentru aceștia:

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

Exercițiul 2. Având următoarea bază de cunoștințe în Prolog, scrisă respectând sintaxa Prolog:

opuse(rosu, violet).

opuse(X, Y) :- opuse(Y, X).

vecine(rosu, portocaliu).

vecine(portocaliu, galben).

vecine(galben, verde).

vecine(verde, albastru).

vecine(X, Y) :- vecine(Y, X).

complementare(rosu, verde).

complementare(violet, galben).

complementare(albastru, portocaliu).

complementare(X, Y) :- complementare(Y, X).

să se scrie un subarbore al arborelui de derivare prin rezoluție SLD pentru următoarea interogare:

?- *complementare(X, Y), complementare(U, V), vecine(X, U), vecine(Y, V).*

cu proprietatea că frunzele etichetate cu \square din acest subarbore dau toate soluțiile acestei interogări.

Adică, în arborele de derivare pentru această interogare, să se expandeze doar suficiente noduri pentru a obține toate soluțiile interogării.

Dacă doriți să expandați mai mult din acest arbore, pentru a arăta structura arborelui, este foarte bine, dar nu este obligatoriu.

Exercițiul 3. Să se scrie în Prolog un predicat binar *termcomplist(Lista, ListaElemListe)*, definit ca mai jos, precum și toate predicatele auxiliare necesare pentru definirea acestuia:

termcomplist să fie satisfăcut dacă ambele sale argumente sunt liste, iar al doilea argument se obține din primul prin înlocuirea elementelor sale care sunt termeni compuși (adică nu sunt nici variabile, nici constante) cu listele formate argumentele operatorilor dominanți ai acestor termeni compuși;

și, într-o interogare în Prolog, *termcomplist* să funcționeze sub forma: dacă primește o listă arbitrară *Listă* în primul argument, să construiască în al doilea argument lista *L* obținută din *Listă* prin înlocuirea elementelor lui *Listă* care sunt termeni compuși cu listele argumentelor operatorilor dominanți ai acestor termeni compuși:

la interogările următoare:	Prologul să răspundă:
?- <i>termcomplist([], L).</i>	$L = []$;
?- <i>termcomplist([1, a, [], X], L).</i>	$L = [1, a, [], X]$;
?- <i>termcomplist([f(X), 1, a, g(a, V), [], X], L).</i>	$L = [[X], 1, a, [a, V], [], X]$;

iar, la interogarea:

?- *termcomplist([1, 2, 3], a, 10, f(1, 2, 3), [X, Y], f(g(1, 2), h(h(c))), f(U, V, h(a))), [c], L).*

Prologul să răspundă:

$L = [[1, [2, 3]], a, 10, [1, 2, 3], [X, [Y]], [g(1, 2), h(h(c))), f(U, V, h(a))], [c, []]]$.

Exercițiul 4. Să se scrie în Prolog un predicat de aritate 4

varterm(Termen1, Termen2, Termen1Modificat, Termen2Modificat)

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

varterm să fie satisfăcut toate cele 4 argumente ale sale sunt termeni Prolog, iar al treilea argument al său se obține din primul prin înlocuirea fiecărei variabile din acesta cu termenul din al doilea argument, în timp ce al patrulea argument al său se obține din al doilea prin înlocuirea fiecărei variabile din acesta cu termenul din primul argument;

și, într-o interogare în Prolog, *varterm* să funcționeze sub forma: dacă primește în primul argument un termen Prolog arbitrar $T1$ și în al doilea argument un termen Prolog arbitrar $T2$, să construiască în al treilea argument termenul obținut din $T1$ prin înlocuirea fiecărei variabile cu $T2$, iar în al patrulea argument termenul obținut din $T2$ prin înlocuirea fiecărei variabile cu $T1$; de exemplu:

la interogările următoare:	Prologul să răspundă:
?- <i>varterm</i> ($a, f(b), T, U$).	$T = a, U = f(b)$;
?- <i>varterm</i> (X, Y, T, U).	$T = Y, U = X$;
?- <i>varterm</i> (X, c, T, U).	$T = c, U = c$;
?- <i>varterm</i> ($g(a, X), f(b), T, U$).	$T = g(a, f(b)), U = f(b)$;
?- <i>varterm</i> ($g(f(V), X), f(g(f(c), f(X))), T, U$).	$T = g(f(f(g(f(c), f(X)))), f(g(f(c), f(X)))),$ $U = f(g(f(c), f(g(f(V), X))))$.