Programare Logică – listă subiecte de examen

Claudia MUREŞAN, c.muresan@yahoo.com, cmuresan@fmi.unibuc.ro
Universitatea din Bucureşti, Facultatea de Matematică şi Informatică

2019–2020, Semestrul I

Exerciţiul 1. Considerăm un limbaj de ordinul I conţinând un simbol de operaţie binară f, unul de operaţie unară g şi un simbol de constantă c. Fie V, X, Y şi Z variabile distincte.

Să se deseneze arborii de expresii asociaţi următorilor doi termeni, apoi, prin aplicarea algoritmului de unificare, să se determine dacă aceşti termeni au unificator şi, în caz afirmativ, să se determine un cel mai general unificator pentru aceştia:

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

Exerciţiul 2. Având următoarea bază de cunoştinţe în Prolog, scrisă respectând sintaxa Prolog:
$este(andrei, activ)$. $este(elena, calm)$. $este(radu, vesel)$.
$iubeste(radu, hamsterii)$.
$iubeste(Persoana, cainii)$:- $este(Persoana, activ)$.
$iubeste(Persoana, pisicile)$:- $este(Persoana, calm)$.
$iubeste(ana, Animal)$:- $iubeste(andrei, Animal)$.
$iubeste(andrei, Animal)$:- $iubeste(elena, Animal)$.
$iubeste(Persoana, Animal)$:- $iubeste(Prieten, Animal)$, $este(Prieten, vesel)$.
să se scrie arborele de derivare prin rezoluţie SLD pentru următoarea interogare:
?- $iubeste(Cine, hamsterii)$.

Exerciţiul 3. Să se scrie în Prolog un predicat binar

$$elimdupconsec(ListaListe, ListaListeFaraDuplicatePePozitiiConsecutive),$$

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

elimdupconsec să fie satisfăcut ddacă ambele argumente ale sale sunt liste de liste, iar al doilea argument al său se obţine din primul său argument $ListL$ prin eliminarea din fiecare listă L aparţinând lui $ListL$ a duplicatelor de pe poziţii consecutive, mai precis: dacă un set de termeni literal identici se află pe (oricâte) poziţii consecutive în lista L, atunci se va păstra în lista care va înlocui pe L o singură copie a acelui termen;

şi, într–o interogare în Prolog, *elimdupconsec* să funcţioneze sub forma: dacă primeşte o listă arbitrară de liste $ListL$ în primul argument, să obţină în al doilea argument lista elementelor lui $ListL$ din care se elimină duplicatele de pe poziţii consecutive, mai precis se elimină toate apariţiile de termeni literal identici de pe poziţii consecutive mai puţin câte o singură apariţie a acestor termeni; de exemplu:

la interogările următoare:	Prologul să răspundă:
?- $elimdupconsec([\,], Lista)$.	$Lista = [\,]$;
?- $elimdupconsec([[\,]], Lista)$.	$Lista = [[\,]]$;
?- $elimdupconsec([[1, 1.0, 0, 0.0], [a, b, c, a], [a, b, a, b]], Lista)$.	$Lista = [[1, 1.0, 0, 0.0], [a, b, c, a], [a, b, a, b]]$;
?- $elimdupconsec([[A, a, b, a, b, B], [X, X, V, X, c, c, c, c, c]], Lista)$.	$Lista = [[A, a, b, a, b, B], [X, V, X, c]]$.
?- $elimdupconsec([[1, 1, f(X), f(X), f(a), f(X), f(V), 1]], Lista)$.	$Lista = [[1, f(X), f(a), f(X), f(V), 1]]$.

Exerciţiul 4. Să se scrie în Prolog un predicat unar $difarunit(Termen)$ definit ca mai jos, precum şi toate predicatele auxiliare necesare pentru definirea acestuia:

$difarunit$ să fie satisfăcut ddacă argumentul său este un termen Prolog cu proprietatea că toţi operatorii f care apar în acest termen au drept argumente numai variabile sau termeni ai căror operatori dominanţi au aritatea mai mică decât aritatea lui f cu exact o unitate, adică toate argumentele lui f sunt fie variabile, fie au operatorii dominanţi de aritate egală cu aritatea lui f minus 1; de exemplu:

la interogările următoare:	Prologul să răspundă:
?- $difarunit(X)$.	$true$;
?- $difarunit(c)$.	$true$;
?- $difarunit(h(h(c)))$.	$false$;
?- $difarunit(f(h(c), f(a, b)))$.	$false$;
?- $difarunit(f(a, X))$	$false$;
?- $difarunit(f(a, f))$	$false$;
?- $difarunit(f(X, Y))$	$true$;
?- $difarunit(f(g(h(a), h(X)), V, k(X, h(c))))$	$true$.