

## Examen

P1	P2	P3	P4	P5	Of	Total
					1	

(P1) [15 pct.] Aplicați algoritmul de unificare din curs pentru termenii

$$f(x, g(x), g(a), v) \text{ și } f(h(y, v), g(h(z, u)), z, g(z)),$$

unde  $x, y, z, u, v$  sunt variabile,  $a$  este simbol de constantă,  $g$  este simbol de operație de aritate 1,  $h$  este simbol de operație de aritate 2 iar  $f$  este simbol de operație de aritate 3.

(P2) [25 pct.] Fie  $\mathcal{L}$  un limbaj de ordinul I cu  $\mathbf{C} = \{a\}$ ,  $\mathbf{F} = \{g\}$ ,  $\mathbf{R} = \{p\}$  unde  $ar(g) = ar(p) = 1$ .

(a) [5 pct.] Determinați forma prenex și forma Skolem pentru formula  $\neg(\forall x p(g(x))) \rightarrow p(g(a))$

(b) [10 pct.] Determinați universul Herbrand determinați expansiunea Herbrand pentru formula de la punctul (a).

(c) [10 pct.] Cercetați dacă formula de la punctul (a) este satisfiabilă (justificând răspunsul dat).

(P3) [30 pct] Fie KB următoarea bază de cunoștințe definită în Prolog:

```
mam(X) :- cal(X).
mam(X) :- pisica(X).
ovi(X) :- rata(X).
ovi(X) :- gasca(X).
ver(X) :- mam(X).
ver(X) :- ovi(X).
pisica(cicero). cal(tulli). rata(ovidiu). gasca(vergi).
```

(a) Determinați universul Herbrand și baza Herbrand asociate acestui program.

(b) Găsiți o respingere SLD pentru întrebarea:

$? - \text{ver}(\text{tulli}).$

## Laborator

### Subiectul I[10 pct.]

Definiți un predicat `list2set/2` care “transformă” o listă într-o mulțime.

```
?- list2set([casa, 2, 3 ,casa, pisica, 3 ,4], Set).  
Set = [casa, 2, 3, pisica, 4].
```

### Subiectul II[20 pct.]

În această problemă vom lucra cu grafuri **orientate**. Un arc va fi definit prin predicatul `arc(a,b)`. (arcul este orientat, poate fi parcurs numai de la primul element la al doilea). Un graf va fi o bază de cunoștințe, de exemplu:

```
arc(a,b).  
arc(b,c).  
arc(b,d).  
arc(d,e).  
arc(a,f).
```

(a) Presupunând că graful nu are cicluri, scrieți un predicat `drum(X,Y)` care să întoarcă `true` dacă există un drum de la `X` la `Y` (și `false` în caz contrar).

```
?- drum(a,e).  
true  
?- drum(c,e).  
false
```

(b) Presupunând că graful poate avea cicluri, scrieți un predicat `drum(X,Y,Drum)` astfel încât, pentru `X` și `Y` date, variabila `Drum` să fie instanțiată cu un drum fără cicluri de la `X` la `Y` (și să întoarcă `false` dacă nu există un drum).

```
?- drum(a,e,Drum).  
Drum=[a,b,d,e].  
?- drum(c,e,Drum).  
false
```