

Seminar 4

Unificare. Programare în Prolog

(S4.1) Folosind algoritmul de unificare, indicați dacă următoarea listă de termeni are unificator:

$$U = p(x, f(y, z)), p(x, a), p(x, g(h(k(x))))$$

(p, f sunt simboluri de funcții de aritate 2, g, h, k sunt simboluri de funcții de aritate 1, a este simbol de constantă, x, y, z sunt variabile).

(S4.2) Găsiți răspunsul dat de Prolog la următoarele întrebări:

1. 'Luke' = luke.
2. Luke = luke.
3. jedi(luke) = luke.
4. jedi(luke) = X.
5. jedi(luke) = jedi(X).
6. father(vader, A) = father(B, luke).
7. heroes(han, X, luke) = heroes(Y, ben, X).
8. heroes(han, X, luke) = heroes(Y, ben).
9. jedi(X) = X.
10. characters(hero(luke), villain(vader)) = characters(X, Y).
11. characters(hero(luke), X) = characters(X, villain(vader))

(S4.3) În această problemă W,R,O,N,G,I,H,T reprezintă niște cifre distincte de la 0 la 9. Scrieți un program în Prolog care găsește valori pentru aceste variabile astfel încât expresia

$$\text{WRONG} + \text{WRONG} = \text{RIGHT}$$

să fie adevărată (unde WRONG și RIGHT sunt numere cu câte 5 cifre).

De exemplu, dacă $W = 1, R = 2, O = 7, N = 3, G = 4, I = 5, H = 6, T = 8$ atunci avem
 $12734 + 12734 = 25468$

(S4.4) Presupunem că avem următoarea bază de cunoștințe:

```

q(a).
q(b).
r(c).
r(d).
s(e).
top(X,Y) :- p(X,Y).
top(X,X) :- s(X).
p(X,Y) :- q(X), r(Y).
p(X,Y) :- s(X), r(Y).

```

Desenați arborele de execuție pentru întrebarea

```
?- top(X,Y).
```

indicând răspunsul pentru fiecare ramură.

(S4.5) Scrieți în Prolog un predicat `list_poz(L,R)` care este adevărat dacă `R` este o listă de perechi care conțin elementele din `L` și poziția pe care se află în `L`, pozițiile fiind în ordine crescătoare:

```
list_poz([a,b,c], [(1,a),(2,b),(3,c)])
```

Predicatul trebuie să verifice, dar și să instanțieze.

(S4.6) Implementați algoritmul *Quicksort* în Prolog.

(S4.7) (1)Scrieți un predicat `flatten(L,R)` unde `L` este o listă de liste, iar `R` conține elementele lui `L`, în aceeași ordine, dar elimină gruparea în liste:

```
flatten([[1,2], [3,4], [5]], [1,2,3,4,5]).
```

Predicatul trebuie să facă verificare și să găsească `R` cu `L` instanțiat.