

# Laboratorul 3 - Programare Logică și Funcțională

Seria 36

Martie 2024

În acest laborator, vom implementa în Prolog formulele propoziționale și semantica lor.

Variabilele vor fi reprezentate de atomi în Prolog (dacă `atom(X)` este `true`, atunci  $X \in Var$ ). Operatorii pe care îi vom utiliza sunt  $\neg$ ,  $\wedge$ ,  $\vee$  și  $\rightarrow$ , pe care îi vom implementa individual, spre deosebire de seminar, iar ei vor fi reprezentați în Prolog prin `non/1`, `and/2`, `or/2`, respectiv `imp/2`.

Spre exemplu, formula logică  $p \rightarrow (q \vee r)$  va fi în reprezentarea noastră:

```
?- X = imp(p, or(q, r)).
```

Scopul laboratorului va fi determinarea algoritmică a faptului că o formulă este sau nu tautologie.

## 1 Exercițiul 1

Definiți un predicat `vars/2` care este adevărat exact atunci când primul argument este o formulă, iar al doilea argument este lista care reprezintă mulțimea variabilelor care apar în ea.

```
?- vars(imp(non(a), imp(a, b)), S).  
S = [a, b]
```

Utilizați, pentru implementare, predicatul predefinit `atom/1`, respectiv predicatul `union/3`, care calculează reuniunea a două liste, considerate ca fiind mulțimi.

## 2 Exercițiul 2

Vom considera evaluările  $e : Var \rightarrow \{0, 1\}$  ca fiind reprezentate printr-o listă de perechi de forma `[(a, 1), (b, 0)]` (în evaluarea curentă,  $e(a) = 1$  și  $e(b) = 0$ ). Definiți un predicat `val/3` astfel încât pentru orice variabilă  $V$  și orice evaluare  $E$  să avem că `val(V, E, A)` este adevărat exact atunci când  $A$  este  $E(V)$ .

```
?- val(b, [(a, 1), (b, 0)], X).  
X = 0
```

## 3 Exercițiul 3

Definiți predicate `bnon/2`, `band/3`, `bor/3`, `bimp/3` care implementează operațiile  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$  pe mulțimea  $\{0, 1\}$ .

```
?- band(1, 0, C).  
C = 0
```

```
?- bimp(A, 0, 0).  
A = 1
```

```
?- bimp(0, B, 0).  
false
```

Puteți defini unele operații în funcție de altele.

## 4 Exercițiul 4

Definiți un predicat `eval/3` astfel încât pentru orice formulă `X` și orice evaluare `E`, avem că pentru orice `A`, `eval(X, E, A)` este adevărat exact atunci când `A` este  $E^+(X)$ .

```
?- eval(imp(b,d), [(a, 1), (b, 0), (d, 1)], A).  
A = 1
```

```
?- eval(imp(d,b), [(a, 1), (b, 0), (d, 1)], A).  
A = 0
```

## 5 Exercițiul 5

Definiți un predicat `evals/3` astfel încât, pentru orice formulă `X` și orice listă de evaluări `Es`, avem că pentru orice `As`, `evals(X, Es, As)` este adevărat atunci când `As` este lista rezultatelor evaluării lui `X` în fiecare dintre elementele lui `Es`.

```
?- evals(imp(d, b), [[(a, 1), (b, 0), (d, 1)], [(a, 1), (b, 1), (d, 0)]], As).  
As = [0, 1]
```

## 6 Exercițiul 6

Definiți un predicat `evs/2` astfel încât, pentru orice listă de variabile `S`, avem că pentru orice `Es`, `evs(S, Es)` este adevărat exact atunci când `Es` este lista evaluărilor definite pe `S`.

```
?- evs([c, b], Es).  
Es = [[(c, 0), (b, 0)], [(c, 1), (b, 0)], [(c, 0), (b, 1)], [(c, 1), (b, 1)]]
```

## 7 Exercițiul 7

Definiți un predicat `all_evals/2` astfel încât, pentru orice formulă `X`, avem că pentru orice `As`, `all_evals(X, As)` este adevărat exact atunci când `As` este lista rezultatelor evaluării lui `X` în fiecare dintre elementele listei evaluărilor definite pe variabilele din `X`.

```
?- all_evals(imp(a, a), As).  
As = [1, 1]
```

```
?- all_evals(imp(a, b), As).  
As = [1, 0, 1, 1]
```

## 8 Exercițiul 8

Definiți un predicat `taut/1` astfel încât, pentru orice formulă `X`, avem că `taut(X)` este `true` exact atunci când `X` este tautologie.

```
?- taut(imp(a, a)).  
true
```

```
?- taut(imp(a, b)).  
false
```