

LABORATOR 5:

! Scop = implementarea funcției în Prolog care ne va ajuta să determinăm dacă o formulă este sau nu tautologie

1.

primul arg = formula,

al doilea = lista cu mulțimea var care apar în formulă
 și returnează "lista formată din var V"

$\text{vars}(V, [V]) :- \text{atom}(V).$

↳ fct. din Prolog care verifică dacă e atom
 ↳ ce atomi am în V

$\text{vars}(S, [X, Y, S]) :- \text{vars}(X, T), \text{vars}(Y, U), \text{union}(T, U, S).$

↳ ce atomi am în X

↳ fct. care unește lista T cu lista U

↳ la fel scris și pt. $\text{sau}(X, Y), \text{non}(X, Y), \text{imp}(X, Y)$

2.

$\text{val}(b, [a, 1], (b, 0), J, A).$!! când facem evaluările pun literă mare pt. a mare unde erau substituie

$A = 0$

↳ mai aici avem evaluările unei variabile

și vrem ca A să aibă evaluarea argumentului dat pe poz. 1

$\text{val}(V, [V, A], J, A).$ ①

$\text{val}(V, [V, T], A) :- \text{val}(V, T, A).$ ②

explicații:

intră întâi pe ① și dacă headul listei are V pe prima poz. atunci pune A ca ultim argument.

dacă ① e fals intră pe ② unde știm sigur că headul nu este ceea ce căutăm așa că reape- lăm pt. Tail.

3.

$bnon / bsi / bsau / bimp$

↳ se referă la val. booleană

$bnon(1, 0), bnon(0, 1).$

$bsi(0, 0, 0), bsi(0, 1, 0), bsi(1, 0, 0), bsi(1, 1, 1).$

$bsau(1, 1, 1), bsau(1, 0, 1), bsau(0, 1, 1), bsau(0, 0, 0).$

$bimp(X, Y, Z) :- bnon(X, NX), bsau(NX, Y, Z).$

↳ știm că $a \rightarrow b = \neg a \vee b$ ↳ $\neg X \wedge X \vee Y$ și punem în Z val.

4.

$\text{eval}(\text{imp}(b, d), [a, 1], (b, 0), (d, 1), J, A).$

$A = 1$

mai trebuie să luăm $e(b), e(d)$ din listă și să punem rez $\text{imp}(e(b), e(d))$ în A

$eval(imp(X,Y), L, A) :- val(X, L, VX), val(Y, L, VY), bimpl(VX, VY, A).$

ne vom folosi de ex 2 și vom lua din listă pe rând valorile lui X și Y în VX și VY și după ca ex 3. vom pune în A val. implicatelor

5. $evals(imp(d,b), [[0,1], [1,0], [1,1], [1,0], [1,0]], As).$
 $As = [0,1]$
 mai multe evaluări

Sequel acestui predicat este să creeze o listă cu evaluările formulei $imp(d,b)$ / sau $(d,b) \dots$ (în cazul asta $imp(d,b)$), preluând evaluările variabilelor din lista de liste

- va. din lab

$evals(-, [], []).$ + cazul de bază

$evals(X, [E|Es], [A|As]) :- eval(X, E, A), evals(X, Es, As).$

- va. pe care am scris-o eu

$evals(-, [], []).$

$evals(imp(X,Y), [H|T], [A1|A2]) :- evals(imp(X,Y), T, A2), eval(imp(X,Y), H, A1).$
 așa trebuie să și operezi

27.

6. $pos([c,b], Es).$

$Es = [[c,0], [b,0], [c,1], [b,0], [c,0], [b,1], [c,1], [b,1]]$

Sequel acestui predicat este să returneze lista tuturor evaluărilor posibile pt c și b
 listă de liste

$pos([], []).$

$pos([V|T], Es) :- pos(T, Esp), adauga(V, Esp, Es).$

adauga(-, [], []).

$adauga(V, [E|T], [[V,0|E], [V,1|E] | Es]) :- adauga(V, T, Es).$

$pos([c,b], Es)$
 Head-ul Tail-ul

1) $\rightarrow pos([c|b], Es) :- pos([b], Esp), adauga(c, Esp, Es)$

2) $\rightarrow pos([b|], Esp) :- pos([], Esp2), adauga(b, Esp2, Esp)$

3) \rightarrow $\cos | [J, [CJ]] |$ \downarrow Esp 2

3) \rightarrow $\text{Esp} = [[5], [[5]]]$, Esp_2

4) \rightarrow $\text{Esp}_2 = [[5]]$ și cont. cu adăugarea lui 6, $[[5]]$, Esp_1
 me întorc la 1