

Aplicații Prolog privind sortarea listelor

- 1) Fie L o mulțime de liste cu valori reale, $[L] = \{ \overset{HL}{\underbrace{L_1, L_2, \dots, L_n}_{TL}} \}$ cu $L_i = [H_i^1, H_i^2, \dots, H_i^{n_i}]$, $H_j^i \in \mathbb{R}$, $i = 1..n$, n_i = numărul de elemente din lista L_i .
- (a) să se determine lista elementelor comune din listele L_i , $i = 1..n$
 $LR = L_1 \cap L_2 \cap \dots \cap L_n$ și să se sorteze lista obținută.
- (b) să se sorteze lista obținută prin concatenarea celor n liste din L .

Program Prolog

domains

lista = real*

lista_liste = lista*

predicates

intersectie(lista, lista, lista)

member(real, lista)

intersectieH(lista_liste, lista)

qsort(lista, lista)

split(real, lista, lista, lista)

concatenare(lista, lista, lista)

insertsort(lista, lista)

insert(real, lista, lista)

sort_lista_intersectie(lista_liste, lista)

sort_lista_concatenare(lista_liste, lista)

clauses

member(X, [X|_]) : !.

member(X, [_|T]) :- member(X, T).

intersectie([], L2, []).

intersectie([H|_], L2, [H|TR]) :- member(H, L2),
intersectie(T, L2, TR), !.

intersectie([H|_], L2, R) :- intersectie(T, L2, R).

intersectieH([], []).

intersectieH([HL|TL], LI) :- intersectieH(TL, LI1),
intersectie(HL, LI1, LI).

qsort([], []).

qsort([H|_], LS) :- split(H, T, U, L2), qsort(L1, LS1),
qsort(L2, LS2), concatenare(LS1, [H|LS2], LS).

split(K, [], [], []).

split(K, [H|_], [H|T1], L2) :- H < K, split(K, T, T0, L2), !.

split(K, [H|_], L1, [H|T2]) :- split(K, T, L1, T2).

concatenare([], L2, L2).

concatenare([H|_], L2, [H|TR]) :- concatenare(T, L2, TR).

sort_list_introsort(LL, LIS): ~ introsort(LL, LI), $\text{gsort}(LI, LIS)$.