**I Laborator 1 PLF P1**

**II** **Liste în Prolog (I)**

**III Predat la:** 14.10.2021

**IV Problema 9**

1. Să se scrie un predicat care întoarce intersecția a două mulțimi.
2. Să se construiască lista *(m, ..., n)*, adică mulțimea numerelor întregi din intervalul

*[m, n]*.

**V Model matematic recursiv**

1. Intersecția a două mulțimi

Având două mulțimi:

A = { a1 , a2, ..., an }

B = { b1 , b2, ..., bm }, cu a1 , a2, ..., an , b1 , b2, ..., bm ∈ N

pentru a determina intersecția acestora, putem doar să luăm elementele dintr-o singură mulțime și să verificăm, pe rând, dacă se regăsesc în cealaltă mulțime. Ne folosim de o funcție de căutare care să verifice dacă un element din A se regăsește în B, sau invers.

Model matematic:

Convenție: a1 , a2, ..., an este echivalent cu mulțimea L1, iar b1 , b2, ..., bm este echivalent cu mulțimea L2.

intersectie(a1 , a2, ..., an , b1 , b2, ..., bm) =

intersectie(L1, L2)=

cauta(a1 , b1 , b2, ..., bm) = cauta(a1, L2) =

=

1. Pentru a converti un interval intr-o listă cu numere întregi, începem cu capătul inferior *m* și parcugem intervalul până ajungem la capătul superior *n*.

Model matematic:

conversie(m, n) =

**VI Cod sursă**

1. %Intersectia a doua multimi

domains

multime=integer\*

predicates

intersectie(multime, multime, multime) %(i, i, o)

cauta(integer, multime) %(i, i)

clauses

cauta(E, [E|\_]).

cauta(E, [\_|T]):- cauta(E, T).

intersectie([], \_, []).

intersectie([X|M1], M, [X|M2]):- cauta(X, M),

intersectie(M1, M, M2).

intersectie([X|M1], M, M2):- not(cauta(X, M)),

intersectie(M1,M,M2).

1. %Lista din interval

domains

lista=integer\*

predicates

conversie(integer,integer,lista)

clauses

conversie(M,N,L) :- M>N, L=[].

conversie(M,N,L) :- M=N, L=[M].

conversie(M,N,[H|T]) :- M<N, H is M, A is M+1,

conversie(A,N,T).

**VII Exemple de testare**

Cazurile sunt scrise de forma(date de intrare=>rezultat).

1. **CAZUL 1**(prima ramură)

A = , B = => intersectie(a1, b1) =

**CAZUL 2**(a doua ramură: verifică primul element, este valid și trece la următorul)

A = {1}, B = {1} => intersectie(1, 1) = {1}

A = {1}, B ={1, 2} => intersectie(1, 1, 2) = {1}

**CAZUL 3**(a treia ramură: primul element nu este valid, îl caută pe următorul ș.a.m.d până termină o mulțime)

A = {1, 2, 3, 4}, B = {3, 4, 5, 6, 7} => intersectie(1, 2, 3, 4, 3, 4, 5, 6, 7) = {3, 4}

1. **CAZUL 1**(nu ne e dat un interval, așa că returnăm numărul)

m = 5, n = 5 => conversie(5, 5) = 5

**CAZUL 2**( parcurgem intervalul)

m = 2, n = 6 => conversie(2, 6) = (2, 3, 4, 5, 6)