Laborator 4

- LISTE. PREDICATE PREDEFINITE

Putem colecta intr-o lista elemente ce satisfac o anumita proprietate cu ajutorul a 3 metapredicate:

1) bagof(X,P,L)
    a. X este o variabila, pe care noi vrem sa o instantiem. Pentru aceasta, ea trebuie sa apara in interiorul conditiei P.
    b. P este conditia pe care noi vrem sa o satisfaca valorile lui X.
    c. L este lista de valori ale lui X.

2) setof(X,P,L) la fel ca bagof, dar elimina duplicatele, iar lista rezultata este sortata

3) findall(X,P,L) daca nu exista nici un element care sa satisfaca P rezultatul, L, va fi o lista vida.

Diferenta dintre bagof, setof si findall este aceea ca, daca in conditiile lor, P, apar si alte variabile decat X, sa zicem Y, atunci bagof si setof vor intoarce o lista L pentru fiecare valoare a lui Y in parte (bagof va intoarce L cu duplicate, setof fara duplicate). Findall nu va tine cont de restul variabilelor din conditie si va intoarce toate solutiile.

- EXEMPLE

Daca dorim sa obtinem toate solutiile pentru interogarea parinte(X,maria), folosim bagof sau setof sau findall:

```
parinte(ion,maria).
parinte(ana,maria).
parinte(ana,maria).
parinte(maria,elena).
parinte(maria,radu).
parinte(elena,nicu).
parinte(radu,gigi).
parinte(radu,dragos).

%----------------------
?- bagof(X ,parinte(X,maria), L).
L = [ion,ana,ana] ? ;
```

no % nu elimina duplicatele
%----------

?- bagof(X,parinte(X,Y),L).

L = [radu],
Y = dragos ? ;
L = [maria],
Y = elena ? ;
L = [radu],
Y = gigi ? ;
L = [ion,ana,ana],
Y = maria ? ;
L = [elena],
Y = nicu ? ;
L = [maria],
Y = radu ? ;
no
%----------

?- bagof(X,Y^parinte(X,Y),L). % punand Y^ in conditie, i-am spus ca vrem sa nu mai grupeze solutiile in functie de Y
L = [ion,ana,ana,maria,maria,elena,radu,radu] ?;
no
%----------

?- setof(X, parinte(X,maria), L).
L = [ana,ion] ? ;
no % elimina duplicatele
%----------

?- setof(X,parinte(X,Y),L).
L = [radu],
Y = dragos ? ;
L = [maria],
Y = elena ? ;
L = [radu],
Y = gigi ? ;
L = [ana,ion],
Y = maria ? ;
L = [elena],
Y = nicu ? ;
L = [maria],
Y = radu ? ;
%----------

?- setof(X,Y^parinte(X,Y),L).

L = [ana,elena,ion,maria,radu] ? ;
no
%----------------------

?- findall(X,parinte(X,maria),L).
L = [ion,ana,ana] ? ;
no % nu elimina duplicatele
%----------------------

findall(X,parinte(X,Y),L).
L = [ion,ana,ana,maria,maria,elena,radu,radu] ?;
no

Alte predicate predefinite pentru operatii cu liste pot fi gasite aici:

https://sicstus.sics.se/sicstus/docs/4.0.2/html/sicstus/lib_002dlists.html

Daca folositi un predicat predefinit dintre cele prezentate la adresa de mai sus, includeti urmatoarea linie de cod in fisierul in care lucrati:

```
?- use_module(library(lists)).
```

❖ EXERCITII

I.

1. Data fiind o lista, L, care are elemente duplicate, creati un predicat cu care sa obtineti o alta lista, LR, care sa contina pozitiile unui element dat. Indexarea incepe de la 1.
   Exemplu: L= [1 2 3 1 3 4 5 3]. Pentru elementul 3: LR=[3,5,8].
2. Data fiind o lista L cu elemente pe care vrem sa le afisam si o lista, LP, de pozitii din L, scrieti un predicat cu care sa afisati elementele din L de la pozitiile specificate in LP.
   Exemplu: L= [1 2 3 1 3 4 5 3]. LP = [2,4,7] => 2, 1, 5.

II. Date fiind faptele din fisierul bazaFacultate.pl, scrieti reguli cu care sa extrageti informatiile 1-7.
Observatie: Pentru a rezolva cerintele de mai jos, va puteti folosi de metapredicatele setof si findall, precum si predicatele scriese de voi in laboratorul anterior, dar si cele din setul I de exercitii.

1. media notelor unui student dat
2. media notelor de la o anumita materie
3. media notelor pentru o facultate
4. facultatea care are cea mai mare medie a notelor studentilor
5. studentul care a luat cea mai mare nota la o materie data.
6. top N al studentilor, in functie de note, la o materie data
7. pentru o facultate data, sa se afiseze restantierii pentru fiecare materie in parte