Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrare de laborator Nr.2

Disciplina:Sisteme bazate pe cunoștințe

Tema:Interpretarea semantică a propoziţiilor.

A efectuat: st.gr. IA-182 , Ulmanu Cristian

A verificat : lect.univ. Bobicev Victoria

Chișinău 2020

**Sarcina:** De efectuat un program în Prolog care execută interpretarea semantică penru propoziţiile date conform variantei.

**Limbajul prolog**

Prolog (engleză. Prolog) este un limbaj și un sistem de programare logică bazat pe limbajul predictiv al logicii matematice a disjunctului Horn, care este un subset al logicii predicatelor de primă ordine.

Limba este centrată în jurul unui mic set de mecanisme de bază, inclusiv maparea la un eșantion, reprezentarea arborescentă a structurilor de date și preluarea automată a randamentelor. Potrivit pentru rezolvarea problemelor în care sunt tratate obiectele (în special obiectele structurate) și relațiile dintre ele. Prologul, datorită caracteristicilor sale, este utilizat în domeniul inteligenței artificiale, al lingvisticii informatice și al programării non-numerice în general. În unele cazuri, punerea în aplicare a simbolice de calcul pe un alt standard de limbi provoacă necesitatea de a crea o cantitate mare de cod complex, în înțelegere, în timp ce punerea în aplicare a acelorași algoritmi în limbajul Prolog oferă un program simplu, ușor de помещающуюся pe o singură pagină.

**Varianta 11 :**

Stiuca a inghitit o rima.

Ursul mininca mure.

**Program prima propoziție:**

s(Lstart,Lfin):-

np(Lstart,Subject,Lrest),

vp(Lrest,Object,Lfin,Semverb),

append(["Subiect"],Subject,Sbj),

append(["Complement"],Object,Obj),

append(Sbj,Obj,Attributes),

append(["Predicat"],[Semverb],Pr),

append(Pr,Attributes,Rezultat),

printlist(Rezultat),!.

np(Lstart,[Semdet,Semnoun|\_],Lfin):-

det(Lstart,Semdet,Lrest),

n(Lrest,Semnoun,Lfin).

np(L1,[Semnoun],L2):-

n(L1,Semnoun,L2).

vp(L1,Lsemnp,L4,[Sempp,Semverb]):-

prep(L1,Sempp,L2),

vt(L2,Semverb,L3),

np(L3,Lsemnp,L4).

vp(L1,Lsemnp,L3,Semverb):-

vt(L1,Semverb,L2),

np(L2,Lsemnp,L3).

append([], List, List).

append([X|L1], List2, [X|L3]) :-

append(L1, List2, L3).

prep(['a'|R],sema,R).

det(['o'|R],semo,R).

n(['Stiuca'|R],semstiuca,R).

n(['rima'|R],semrima,R).

vt(['inghitit'|R],seminghitit,R).

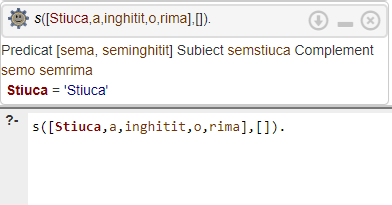
printlist([]).

printlist([X|List]) :-

write(X), write(' '),

printlist(List).

**Execuție :**



**Program a doua propoziție**

s(Lstart,Lfin):-

np(Lstart,Subject,Lrest),

vp(Lrest,Object,Lfin,Semverb),

append(["Subiect"],Subject,Sbj),

append(["Complement"],Object,Obj),

append(Sbj,Obj,Attributes),

append(["Predicat"],[Semverb],Pr),

append(Pr,Attributes,Rezultat),

printlist(Rezultat),!.

np(L1,[Semnoun],L2):-

n(L1,Semnoun,L2).

vp(L1,Lsemnp,L3,Semverb):-

vt(L1,Semverb,L2),

np(L2,Lsemnp,L3).

append([], List, List).

append([X|L1], List2, [X|L3]) :-

append(L1, List2, L3).

n(['Ursul'|R],semurs,R).

n(['mure'|R],semmure,R).

vt(['maninca'|R],semmaninca,R).

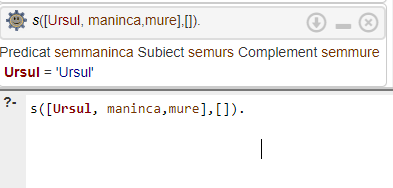
printlist([]).

printlist([X|List]) :-

write(X), write(' '),

printlist(List).

**Execuție :**



**Explicații :** Înainte de execuție introducem propoziția pe care o dorim să o analizeze programa, se introduce fiecare cuvânt sau articol aparte, separat prin virgule. Odată ce am introdus propoziția algoritmul verifică ce parte de vorbire sunt fiecare cuvânt după aceea verifică care este ordinea lor și determină ce cuvinte sunt subiect, predicat sau complement și afișează la ecran rezultatul.