

LUCRAREA DE LABORATOR NR. 1

1. Scop

Insusirea elementelor fundamentale din Teoria Cadrelor (Minsky)

2. Studiu necesar

N. Tandareanu - "Introducere in programarea logica. Limbajul Prolog"

N. Tandareanu - Note de curs

3. Mod de realizare al lucrarii

- Enuntarea problemei de rezolvat
- Descrierea succinta a fisierelor componente
- Codul in Turbo Prolog
- Rezultate obtinute la executie

4. Exemplu

- Enuntul problemei:

Fie urmatoarea baza de cunostiinte:

Animalele respira si mananca regulat.

Masina are motor, 4 roti si consuma in general benzina.

Un lucru atrage atentia si este nostim de obicei.

Omul este un animal cu doua picioare care vorbeste.

Mihai este un om. El are o masina Ford rosie.

Un copil mic este un om care nu vorbeste si nu lucreaza.

Adrian este un copil mic.

Dan este un copil mic care vorbeste.

Un bebelus este un copil mic care atrage atentia si este nostim de obicei.

Andrei este un bebelus care plange totdeauna.

Reprezentati baza de cunostinte folosind Teoria cadrelor. Implementati aceasta reprezentare in Prolog.

- Descrierea fisierelor componente

fișierul kb.pro: va contine predicatele necesare manipularii cunostintelor in vederea interogarii adugarii sau scoaterii lor din baza de cunostinte.

fișierul kb.dat: va contine baza de cunostinte care va fi incarcata pentru interogare.

- Codul Turbo Prolog

fișierul kb.pro:

domains

lista_par=symbol*

slot=slot(symbol,symbol)

lista_sloti=slot*

database

obiect(symbol,lista_par,lista_sloti)

predicates

get_list_s(symbol,symbol,lista_sloti)

```

get_list_p(symbol,lista_par)
afis_ob
list_s(lista_sloti)
scot_list_s(symbol,lista_sloti,lista_sloti)
pun_obiect
membru(symbol,symbol,lista_sloti)
elem_list(lista_par,symbol)
poz(symbol,symbol,lista_par,lista_sloti)
afis_s(symbol)
afis_slot
interog
start
optiune(integer)
scot_obiect
modif_slot
sterg_slot
adaug_slot
menu
sterg_fer(integer)
clauses
/* predicat de start */
start:-makewindow(1,6,7,““,4,0,20,40),
makewindow(2,14,7,““,4,40,20,40),
makewindow(3,3,7,“ OPTIUNE: 1-9 “,0,0,4,80).
/* Meniu */
menu:-sterg_fer(2),sterg_fer(1),
write(“1. Consulta baza de cunoastere\n“),
write(“2. Salveaza baza de cunoastere\n“),
write(“3. Adauga obiect in baza \n“),
write(“4. Sterge obiect din baza \n“),
write(“5. Adauga slot \n“),
write(“6. Sterge slot \n“),
write(“7. Modifica slot \n“),
write(“8. Interogheaza baza \n“),
write(“9. Listeaza sloturi \n“),
write(“0. Iesire program“),
shiftwindow(3),readint(N),N>=1,N<=9,optiune(N),menu.
menu.
/* executa o optiune */
optiune(1):-sterg_fer(3),write(“Consult baza de cunoastere\n“),
retractall(_),consult(“kb.dat“).
optiune(2):-sterg_fer(3),write(“Salvez baza de cunoastere\n“),
save(“baza.dat“).
optiune(3):-sterg_fer(3),write(“Adaug obiect in baza de cunoastere\n“),

```

```

pun_obiect.
optiune(4):-sterg_fer(3),write("Sterg obiect din baza de cunoastere\n"),
scot_obiect.
optiune(5):-sterg_fer(3),write("Adaug slot unui obiect\n"),
adaug_slot.
optiune(6):-sterg_fer(3),write("Sterg slot dintr-un obiect\n"),
sterg_slot.
optiune(7):-sterg_fer(3),write("Modific slot dintr-un obiect\n"),
modif_slot.
optiune(8):-sterg_fer(3),write("Interoghez baza de cunoastere\n"),
interog.
optiune(9):-sterg_fer(3),write("Listez sloturile unui obiect\n"),
afis_slot.
/* 8. interogare */
interog:-sterg_fer(2),sterg_fer(1),write("Lista obiecte:"),
afis_ob,shiftwindow(2),
write("\n\nNume obiect de interogat: "),readln(NO),
write("\nNume data membru: "),readln(DM),
write("\nValoare data membru: "),
obiect(NO,LP,LS),
poz(DM,VDM,LP,LS),
write(VDM),readln(_),interog.
interog.
/* sterge fereastra I */
sterg_fer(I):-shiftwindow(I),clearwindow.
/* pozitionare */
poz(DM,VDM,_,LS):-free(VDM),membru(DM,VDM,LS).
poz(DM,VDM,LP,-):-free(VDM),elem_list(LP,X),obiect(X,LP1,LS1),
poz(DM,VDM,LP1,LS1).
/* verifica daca slot(X,Y) este membru in lista de sloturi */
membru(X,Y,[slot(X,Y)|_]).
membru(X,Y,[_|L]):-membru(X,Y,L).
/* verifica daca lista are elementul X */
elem_list([X|_],X).
elem_list(_|L,X):-elem_list(L,X).
/* afiseaza cadre */
afis_ob:-obiect(NO,_,_),write("\n",NO),fail.
afis_ob.
/* predicate auxiliare */
/* citeste lista sloturi */
get_list_s(S1,S2,[slot(X,Y)|L]):-write("\n",S1),
readln(X),X<>"",
write("\n",S2),readln(Y),Y<>"",
get_list_s(S1,S2,L).

```

```

get_list_s(-,-,[]).
/* citește lista parinti */
get_list_p(S1,[X|L]):-write("\n",S1),
readln(X),X<> "",
get_list_p(S1,L).
get_list_p(-,[]).
/* listează sloturi */
list_s([slot(X,Y)|L]):-write("\n ",X," \t",Y),list_s(L).
list_s([]).
/* afișează lista sloturi pt un obiect */
afis_s(NO):-obiect(NO,-,LS),write("\nSloturi proprii ale obiectului ",NO),
list_s(LS),fail.
afis_s(NO):-obiect(NO,LP,-),elem_list(LP,X),
write("\nobiect parinte"),afis_s(X).
goal
start,menu.

fișierul kb.dat:
obiect("animal",[ ],[slot("respira","da"),slot("mananca","regulat")])
obiect("masina",[ ],[slot("nr_roti","4"),slot("motor","da"),slot("consuma","c(benzina,in_general)")])
obiect("lucru",[ ],[slot("atrage_atentia","da"),slot("nostim","de_obicei")])
obiect("om",[ "animal"],[slot("nr_picioare","2"),slot("vorbeste","da")])
obiect("mihai",[ "om"],[slot("poseda","m1")])
obiect("m1",[ "masina"],[slot("marca","ford"),slot("culoare","rosu")])
obiect("copil_mic",[ "om"],[slot("lucreaza","nu"),slot("vorbeste","nu")])
obiect("adrian",[ "copil_mic"],[])
obiect("dan",[ "copil_mic"],[slot("vorbeste","da")])
obiect("bebelus",[ "copil_mic"],[slot("atrage_atentia","da"),slot("nostim","de_obicei")])
obiect("andrei",[ "bebelus"],[slot("plange","totdeauna")])

/* Teme: Sa se implementeze urmatoarele predicate: */
/* 3. adauga cadru in KB */
pun.obiect.
/* 4. scoate cadru din KB */
scot.obiect.
/* 5. adauga un slot */
adaug_slot.
/* 6. sterge un slot */
sterg_slot.
/* predicat auxiliar -> scot_list_s(NS,LS,LS1) scoate slotul NS din lista de
sloturi LS si returneaza LS1*/
scot_list_s(,).
/* 7. modifica slot valoare */
modif_slot.
/* 9. afisez slot */

```

afis_slot.