

ДОМАШНА ЗАДАЧА 3

ЛОГИЧКО ПРОГРАМИРАЊЕ:

РЕПРЕЗЕНТАЦИЈА НА ЗНАЕЊЕ -ИНТЕГРАМ

Марија Вецовска 185008

1. Факти:

1) Полот на личностите

devojka(mira).

momche(bruno).

momche(teo).

momche(igor).

2)Фактите:

1. Тео седи најлево и јаде сендвич.

levo_od(teo, nikoj).

hrana(teo, sendvich).

2. Мира сака да решава крстозбори и ужива во јадењето пита.

hrana(mira, pita).

hobi(mira, krstozbori).

3. Девојката има бела маица.

maica(X, bela) :- devojka(X).

4. Бруно има жолта маица.

maica(bruno, zholta).

5. Оној што сака да пишува јаде хамбургер.

hrana(X, hamburger) :- hobi(X, pishuvanje).

6. Личноста која јаде пита седи покрај Тео.

sedi_pokraj(X, teo) :- hrana(X, pita).

7. Бруно седи покрај оној што јаде пица.

sedi_pokraj(bruno, X) :- hrana(X, pica).

8. Личноста која седи покрај онаа во бела маица сака пица.

sedi_pokraj(X, Y) :- hrana(X, pica), maica(Y, bela).

9. Игор сака да чита.

hobi(igor, chitanje).

10. Сина маица има личноста која седи десно од девојката.

desno_od(Y, X) :- maica(X, sina), devojka(Y).

3) Предикати за постоењето на хобињата, храната и бојата на маици за кои нема услов како оние погоре:

hrana(_, pica).
hobi(_, fotografija).
hobi(_, pishuvanje).
maica(_, crvena).
maica(_, sina).

4) Предикат кој означува дека секоја личност мора да седи до мношче или девојка:

sedi_pokraj(_, Y) :- momche(Y).
sedi_pokraj(_, Y) :- devojka(Y).

5) ne_sodrzi((X, Y, Z, W, Lik), L) - предикат кој проверува дали вредностите на променливите: X, Y, Z, W, Lik не се содржат во листата L.

ne_sodrzi((X, Y, Z, W, Lik), L) :- not(member(X, L)), not(member(Y, L)),
not(member(Z, L)), not(member(W, L)), not(member(Lik, L)).

6) licnost(X, Y, Z, W, L, K) – предикат кој гради една личност: X-име, Y-храна, Z-хоби, W-маица, L- име на личноста која седи до X.

За X-име, Y-храна, Z-хоби, W-маица бара соодветни записи во базата, и им доделува вредности така што нема да има противречности.

Пример, hrana(X, hamburger) :- hobi(X, pishuvanje) значи дека на тој што ќе му додели храна хамбургер мора да му додели и хоби пишување.

Потоа наоѓа личност кој седи до(поточно десно од) X со предикатот:

(1) desno_od(X, L), L \= X,

(2) sedi_pokraj(X, L), L \= X,

(3) sedi_pokraj(X, L), L \= X,

(2) и (3) го означуваат истото, мора да се проверат двете можни комбинации на запишување на X, L затоа што тие го означуваат истото, и не се знае на кој начин се запишани како факти.

ne_sodrzi((X, Y, Z, W, L), K) – проверка дали вредностите кои се избрани за таа личност не се наоѓаат во L(не се искористени претходно).

licnost(X, Y, Z, W, L, K) :-

hrana(X, Y), hobi(X, Z), maica(X, W),

desno_od(X, L), L \= X,

ne_sodrzi((X, Y, Z, W, L), K).

(1)

licnost(X, Y, Z, W, L, K) :-

hrana(X, Y), hobi(X, Z), maica(X, W),

sedi_pokraj(X, L), L \= X,

ne_sodrzi((X, Y, Z, W, L), K).

(2)

licnost(X, Y, Z, W, L, K) :-

hrana(X, Y), hobi(X, Z), maica(X, W),

sedi_pokraj(L, X), L \= X,

ne_sodrzi((X, Y, Z, W, L), K).

(3)

7) reshenie(L1) -предикат којшто треба да врати како резултат листа L која ќе го содржи целосното решение на проблемот. Еден елемент на ваквата листа претставува четворка од вредности (Ime, Hrana, Hobi, Maica) и се однесува на точна комбинација на име на студент, храна која ја јаде, хоби кое го има и маица која ја носи.

Решението се довива така што се почнува од најлевиот студент и се редат останатиите понатака.

levo_od(X1, nikoj) – го наоѓа во X студентот кој седи најлево.

licnost(X1, Y1, Z1, W1, X2, []) – потоа се наоѓаат останатите вредности за тој студент со предикатот. Празната листа кој се проследува значи дека првиот студент нема рестрикции за кои вредности може да ги има, затоа што ниедна вредност не е искоритена.

append([], [X1, Y1, Z1, W1], K) во K се враќа листа на вредности кои се искоритени од првиот студент и повеќе не смее да се користат.

Личноста која ќе седи до X1 е X2,

licnost(X2, Y2, Z2, W2, X3, K)

Овој процес се повторува и за втората личност на ист начин, освен што таа не смее да ги користи вредностите во листата K – вредностите на личноста 1.

Овој процес продолжува и за двете наредни личности.

reshenie(L1) :-

levo_od(X1, nikoj),

licnost(X1, Y1, Z1, W1, X2, []), append([], [X1, Y1, Z1, W1], K),

licnost(X2, Y2, Z2, W2, X3, K), append(K, [X2, Y2, Z2, W2], K2),

licnost(X3, Y3, Z3, W3, X4, K2), append(K2, [X3, Y3, Z3, W3], K3),

licnost(X4, Y4, Z4, W4, nikoj, K3), !,

L1 = [(X1, Y1, Z1, W1), (X2, Y2, Z2, W2), (X3, Y3, Z3, W3), (X4, Y4, Z4, W4)].