

Домашна 3

Данило Најков 206033

Решението е:

L = [[teo, sendvic, fotografija, crvena], [mira, pita, krstozbor, bela], [igor, pica, citanje, sina], [bruno, hamburger, pisuvanje, zolta]]

За ова најпрвин ги дефинирам општите факти од задачата со предикати за студент, храна, хоби, маица и девојка за да се опише доменот на знаењето.

student(teo).
student(mira).
student(bruno).
student(igor).

hrana(sendvic).
hrana(pita).
hrana(hamburger).
hrana(pica).

hobi(krstozbor).
hobi(pisuvanje).
hobi(citanje).
hobi(fotografija).

maica(bela).
maica(zolta).
maica(sina).
maica(crvena).

devojka(mira).

Потоа тргнав за предикатот **jade(X,Y)** што опишува - човекот X ја јаде храната Y.

Следните два предикати се дадени општо од условите:

jade(teo, sendvic).
jade(mira, pita).

Следниот предикат следи од условот дека тој што јаде пица седи до личноста со бела маица. Предикатот **pozicija(X,Y)** значи X седи лево од Y. Тука е ставено (*pozicija(X, Y)* ; *pozicija(Y, X)*) за да провери или X е лево од Y, или Y е лево од X (според условот - седи до). Cut операторот тука не е задолжителен, но спречува истото решение да се појавува повеќе пати.

```
jade(X, pica) :-  
    student(X),  
    X \= mira,  
    X \= teo,  
    nosi_maica(Y, bela),  
    (pozicija(X, Y) ; pozicija(Y, X)), !.
```

Последно дефинирано е **jade(X, hamburger)**, за останатиот случај кога студентот не е Мира или Тео (бидејќи тие се дефинирани претходно што јадат) и да не јаде пица (дефиниран е предикатот погоре за што треба да е исполнето за да јаде пица)

```
jade(X, hamburger):-  
    student(X),  
    X \= mira,  
    X \= teo,  
    not(jade(X, pica)).
```

Следен е предикатот **ima_hobi(X,Y)** што означува дека човекот X има хоби Y. Следните два факти се дадени како такви во условите на задачата.

```
ima_hobi(mira, krstozbor).  
ima_hobi(igor, citanje).
```

ima_hobi(X, pisuvanje) е предикатот од условот дека тој што сака пишување јаде хамбургер.

```
ima_hobi(X, pisuvanje):-  
    student(X),  
    X \= mira,  
    X \= igor,  
    jade(X, hamburger).
```

Слично како и кај jade, **ima_hobi(X, fotografija)** ги опфаќа условот што не беше фатен од претходните предикати - да сака фотографија. За студентот да сака фотографија, тој не треба да е Мира или Игор, бидејќи тие веќе имаат хоби, и да нема хоби пишување.

```
ima_hobi(X, fotografija):-  
    student(X),  
    X \= mira,  
    X \= igor,  
    not(ima_hobi(X, pisuvanje)).
```

Следен е предикатот **nosi_maica(X,Y)** што значи човекот X носи Y маица.

Дадено е дека Бруно носи жолта маица:

```
nosi_maica(bruno, zolta).
```

Овој предикат е од условот дека девојката носи бела маица:

```
nosi_maica(X, bela) :-  
    student(X),  
    devojka(X).
```

Следниот е од условот дека тој што носи сина маица седи десно од девојката. *pozicija(Y,X)* означува дека X седи десно од Y.

```
nosi_maica(X, sina) :-  
    student(X),  
    X \= bruno,  
    devojka(Y),  
    pozicija(Y,X).
```

За тој што носи маица треба да важи дека не е еден од дефинираните студенти погоре (Бруно), и да не носи жолта, сина, или бела маица.

```
nosi_maica(X, crvena) :-  
    student(X),  
    maica(Y),  
    X \= bruno,  
    Y \= zolta,  
    Y \= bela,  
    Y \= sina,  
    not(nosi_maica(X, sina)),  
    not(nosi_maica(X, bela)).
```

Предикатот **teo_najlevo** излегува од условот дека Тео седи најлево во групата. Прима два студенти X и Y како аргументи и проверува дека Тео не е десно.

```
teo_najlevo(X, Y) :- student(X), student(Y), Y\=teo.
```

pozicija(X,Y) веќе го објаснив погоре, но формално значи дека X и Y седат еден до друг така што X е лево, а Y на десната позиција. Во сите дефиниции се користи предикатот **teo_najlevo** за да се осигура дека Тео не е на десната позиција.

Следниот предикат следи од условот дека тој што седи до Тео, јаде пита. При што има две дефиниции и за лево и за десно од Тео иако е јасно дека Тео е најлево (за да нема изведени факти).

```
pozicija(teo, Y) :-  
    student(Y),  
    teo_najlevo(teo, Y),  
    jade(Y, пита).
```

```
pozicija(X, teo) :-  
    student(X),  
    teo_najlevo(X, teo),  
    jade(X, пита).
```

Овој предикат е според условот дека тој што седи до Бруно јаде пица.

```
pozicija(X, bruno) :-  
    student(X),  
    X \= bruno,  
    teo_najlevo(X, bruno),  
    jade(X, pica).
```

Позледната дефиниција на предикатот е за сите други случаи што не беа опфатени од претходните дефиниции.

```
pozicija(X, Y) :-  
    student(X),  
    student(Y),  
    X \= bruno,  
    Y \= bruno,  
    X \= teo,  
    Y \= teo,  
    teo_najlevo(X, Y),  
    X \= Y.
```

За наоѓање на решението го дефинирам предикатот **reshenie** кое за почеток го зема студентот најлево со **pozicija(X,_)**, па cut операторот за да не бара друго решение за најлев студент. Се повикува **resenie_vnatresno** со првиот студент.

```
reshenie(L) :-  
    pozicija(X,_),  
    !,  
    resenie_vnatresno(X,L).
```

resenie_vnatresno има две дефиниции:

- Првото е ако се стигне на крајот на редот (до најдесниот студент - тогаш **pozicija(X,_) враќа false**), и во дводеминзиалната листа се ставаат информациите за јадење, хоби и маица за овој студент
- Второто е ако не е стигне крајот, така што се проверува дали во листата на студенти од лево кон десно не се случило повторување на некој студент со **nema_povtoruvanje**, и рекурзивно се повикува **resenie_vnatresno**. При излегувањето од рекурзијата се ставаат информациите за студентот во резултантната листа

Овде е присутен и cut операторот, бидејќи без него го враќа истото решение повеќе пати.

```
resenie_vnatresno(X, [[X,Hr,Ho,Ma]]) :-  
    not(pozicija(X,_)),  
    !,  
    jade(X, Hr),  
    ima_hobi(X, Ho),  
    nosi_maica(X, Ma).
```

```
resenie_vnatresno(X, [[X,Hr,Ho,Ma]|L]) :-  
    pozicija(X, Y),
```

```
nema_povtoruvanje(X, Y),  
!,  
resenie_vnatresno(Y, L),  
jade(X, Hr),  
ima_hobi(X, Ho),  
nosi_maica(X, Ma).
```

На крај **nema_povtoruvanje** се враќа наназад низ позициите (од моменталната позиција па налево) и проверува дека не е веќе спомнат студентот Y. Ова е потребно за да осигураме дека секој студент се појавува максимум еднаш.

```
nema_povtoruvanje(X, _) :-  
    not(pozicija(_, X)).  
nema_povtoruvanje(X, Y) :-  
    pozicija(N, X),  
    N \= Y,  
    nema_povtoruvanje(N, Y).
```