

ДОМАШНА ЗАДАЧА 2

ЛОГИЧКО ПРОГРАМИРАЊЕ БАЗИ НА ПОДАТОЦИ ВО PROLOG

Задача 1. (30 поени) Во базата на знаење во PROLOG се сместени факти за фамилии. Постојат факти за лица од следниот облик

```
lice(Sifra,Ime,Prezime,Pol,Data_raganje,Mesto_raganje,Mesto_ziveenje) ,
```

и факти за фамилии од следниот облик

```
familija(Sifra_tatko,Sifra_majka,Lista_sifri_deca)
```

каде `Lista_sifri_deca` е листа во која се содржат шифрите на децата од тој брак.

На пример:

```
lice(1,petko,petkovski,m,datum(1,3,1950),kratovo,skopje) .
lice(2,marija,petkovska,z,datum(30,5,1954),kumanovo,skopje) .
lice(3,ljubica,petkovska,z,datum(29,11,1965),skopje,skopje) .
lice(4,vasil,vasilev,m,datum(8,4,1954),bitola,bitola) .
lice(5,elena,vasileva,z,datum(19,6,1958),resen,bitola) .
lice(6,krste,krstev,m,datum(9,8,1948),veles,veles) .
lice(7,biljana,krsteva,z,datum(13,8,1949),veles,veles) .
lice(8,igor,krstev,m,datum(26,10,1971),veles,skopje) .
lice(9,kristina,krsteva,z,datum(30,5,1974),kumanovo,skopje) .
lice(10,julija,petrova,z,datum(30,5,1978),skopje,skopje) .
lice(11,bosko,petkovski,m,datum(13,11,1981),skopje,skopje) .
lice(12,gjorgji,vasilev,m,datum(15,7,1978),bitola,bitola) .
lice(13,katerina,petkovska,z,datum(11,12,1979),bitola,skopje) .
lice(14,petar,vasilev,m,datum(21,2,1982),skopje,skopje) .
lice(15,andrej,krstev,m,datum(3,8,1998),skopje,skopje) .
lice(16,martina,petkovska,z,datum(5,12,2005),skopje,skopje) .
```

```
familija(1,2,[9,10]).
familija(1,3,[11]).
familija(4,5,[12,13,14]).
familija(6,7,[8]).
familija(8,9,[15]).
familija(11,13,[16]).
```

а) Да се напише предикат во PROLOG `rodeni_razlicen_grad(Kolku)` кој враќа колку лица се родени во град различен од градовите на раѓање на двата родитела. За пример базата на знаење:

```
?- rodeni_razlicen_grad(Kolku).      ќе одговори:

Kolku=3;
```

б) Да се напише предикат во PROLOG `predci(Sifra,L)` кој на влез прима шифра на некое лице и за тоа лице на излез треба да ги врати сите негови предци (нивните шифри) за кои се исполнети следните услови: предокот да биде од **истиот пол** како и лицето и денот на раѓање на предокот да биде најмногу **една недела пред или после** денот на раѓање на лицето (годината е небитна и претпоставете дека сите месеци имаат по 30 денови). За пример базата на знаење:

```
?- predci(8,L).      ќе одговори:

L=[];

?- predci(15,L).      ќе одговори:

L=[6];

?- predci(16,L).      ќе одговори:

L=[13,3];
```

Задача 2. (30 поени) Во базата на знаење во PROLOG се сместени факти за телекомуникациски оператор. Постојат факти за телефонски броеви од следниот облик

`telefon (Broj, Ime, Prezime, Lista_na_pojdovni_povici) ,`

каде што листата на појдовни повици е составена од елементи повик, каде секој повик го има обликот

`povik (Povikan_broj, Traenje) ,`

каде `Traenje` се однесува на вкупното траење во минути на сите појдовни повици кон дадениот број, и факти за СМС пораки од следниот облик

`SMS (Broj_koj_ja_ispraka_porakata, Lista_broeви_koi_ja_dobivaat_porakata)`

На пример:

`telefon (111111, petko, petkovski, [povik (222222, 250) , povik (101010, 125)]) .`

`telefon (222222, marija, petkovska, [povik (111111, 350) , povik (151515, 113) , povik (171717, 122)]) .`

`telefon (333333, ljubica, petkovska, [povik (555555, 150) , povik (101010, 105)]) .`

`telefon (444444, vasil, vaselev, [povik (171717, 750)]) .`

`telefon (555555, elena, vaseleva, [povik (333333, 250) , povik (101010, 225)]) .`

`telefon (666666, krste, krstev, [povik (888888, 75) , povik (111111, 65) , povik (141414, 50) , povik (161616, 111)]) .`

`telefon (777777, biljana, krsteva, [povik (141414, 235)]) .`

`telefon (888888, igor, krstev, [povik (121212, 160) , povik (101010, 225)]) .`

`telefon (999999, kristina, krsteva, [povik (666666, 110) , povik (111111, 112) , povik (222222, 55)]) .`

`telefon (101010, julija, petrova, []) .`

`telefon (121212, bosko, petkovski, [povik (444444, 235)]) .`

`telefon (131313, gjorgji, vaselev, [povik (141414, 125) , povik (777777, 165)]) .`

`telefon (141414, katerina, petkovska, [povik (777777, 315) , povik (131313, 112)]) .`

`telefon (151515, petar, vaselev, []) .`

`telefon (161616, andrej, krstev, [povik (666666, 350) , povik (111111, 175) , povik (222222, 65) , povik (101010, 215)]) .`

`telefon (171717, martina, petkovska, [povik (222222, 150)]) .`

`sms (111111, [222222, 999999, 101010]) .`

`sms (444444, [333333, 121212, 161616]) .`

`sms (111111, [777777]) .`

`sms(666666,[888888]).`

`sms(444444,[555555,121212,131313,141414]).`

`sms(666666,[777777,888888]).`

`sms(888888,[999999,151515]).`

`sms(171717,[131313,161616]).`

- a) Да се напише предикат во PROLOG `najbroj(x,y)` со кој ќе се врати името `x` и презимето `y` за бројот кој остварил комуникација со најмногу други броеви (се земаат само повиците). За пример базата на знаење:

`?-najbroj(X,Y).` ќе одговори:

`X=julija, Y=petrova;`

- б) Да се напише предикат во PROLOG `omilen(x,y)` кој за даден број `x` проследен на влез ќе го врати неговиот омилен број `y`. Омилен број е оној со кој има остварено најголемо вкупно траење на повици (се собираат и појдовни и дојдовни). При пресметувањето на омилениот број секоја СМС порака да се разгледува како остварен повик со траење 100 (оној кој ја испраќа пораката е појдовниот број, оние до кој се испраќа пораката се дојдовни). За пример базата на знаење:

`?-omilen(888888,Y).` ќе одговори:

`Y=666666`

`?-omilen(666666,Y).` ќе одговори:

`Y=161616`

`?-omilen(999999,Y).` ќе одговори:

`Y=111111`

`?-omilen(111111,Y).` ќе одговори:

`Y=222222`

Задача 3. (40 поени) Во базата на знаење на PROLOG се сместени факти за остварени услуги од страна на клиентите на една такси компанија:

klient(klientid, ime, prezime, lista_na_ostvareni_uslugi)

при што првите два аргументи се однесуваат на името и презимето на клиентот, третиот аргумент е листа од услуги кои ги остварил соодветниот клиент (листата има најмалку еден елемент). Листата на остварени услуги е составена од информации за локација од кое тргнува и локација до кое треба да стигне такси возилото, цената по поминат километар, датумот и бројот на такси возилото т.е. структури во облик

usluga(mesto_od, mesto_do, cena_kilometar, datum(den, mesec, godina), broj_vozilo).

На пример:

```
klient(1, petko, petkov, [usluga(a, b, 50, datum(12, 12, 2015), 23), usluga(c, a, 50, datum(7, 12, 2015), 34), usluga(c, f, 40, datum(7, 11, 2015), 23)]).
```

```
klient(2, vasil, vasiliev, [usluga(a, e, 50, datum(25, 12, 2015), 12), usluga(c, g, 40, datum(17, 11, 2015), 56), usluga(g, d, 50, datum(17, 12, 2015), 45), usluga(e, a, 40, datum(24, 12, 2015), 34)]).
```

```
klient(3, krste, krstev, [usluga(c, b, 60, datum(31, 12, 2015), 56), usluga(e, f, 60, datum(31, 12, 2015), 34)]).
```

```
klient(4, petar, petrov, [usluga(a, f, 50, datum(25, 12, 2015), 23), usluga(f, d, 50, datum(25, 12, 2015), 34)]).
```

```
klient(5, ivan, ivanov, [usluga(d, g, 50, datum(7, 12, 2015), 56), usluga(g, e, 40, datum(25, 12, 2015), 34)]).
```

```
klient(6, jovan, jovanov, [usluga(c, f, 50, datum(5, 12, 2015), 12), usluga(f, d, 50, datum(27, 12, 2015), 45)]).
```

```
klient(7, ana, aneva, [usluga(e, d, 50, datum(11, 12, 2015), 12), usluga(d, g, 50, datum(11, 12, 2015), 12)]).
```

```
klient(8, lidiya, lideva, [usluga(e, g, 50, datum(29, 12, 2015), 45), usluga(f, b, 50, datum(29, 12, 2015), 34)]).
```

Доколку постои директна врска помеѓу две локации за тој пар во базата се чува факт за должината на врската (растојанието изразено во километри), во облик *rastojanie(LokA, LokB, Km)*. На пример:

```
rastojanie(a, b, 4).
```

```
rastojanie(a, c, 7).
```

```
rastojanie(b, c, 5).
```

```
rastojanie(b,d,3) .  
rastojanie(c,d,4) .  
rastojanie(b,e,6) .  
rastojanie(c,e,2) .  
rastojanie(b,f,8) .  
rastojanie(e,f,5) .  
rastojanie(f,g,3) .
```

Кога таксистите возат помеѓу две локации ја поминуваат патеката која има минимална должина од сите можни патеки помеѓу локациите.

а) Напишете предикат во PROLOG `izbroj_lokacija(Lok,Br)` кој за локација `Lok` која се задава на влез ќе пресмета колку пати таа локација била почетна или крајна за некоја услуга. За пример базата на знаење:

```
?- izbroj_lokacija(a,Br) .      ќе одговори:  
  
Br=5;  
  
?- izbroj_lokacija(f,Br) .      ќе одговори:  
  
Br=7;
```

б) Напишете предикат во PROLOG `najmnogu_kilometri(X,Y)` кој во `X` и `Y` ќе ги враќа името и презимето на клиентот кој има поминато најмногу километри со такси компанијата. За пример базата на знаење:

```
?- najmnogu_kilometri(X,Y) .    ќе одговори:  
  
X=vasil, Y=vasilev;
```

в) Напишете предикат во PROLOG `najmnogu_zarabotil(X)` со кој ќе го најдете бројот на такси возилото `X` кое заработило најмногу во текот на месец декември 2015 година. За пример базата на знаење:

```
?- najmnogu_zarabotil(X) .      ќе одговори:  
  
X=34;
```

ПРИКАЧУВАЊЕ И ПРОВЕРКА НА ВАШЕТО РЕШЕНИЕ НА ДОМАШНАТА ЗАДАЧА

Како решение за домашната задача треба да креирате две датотеки: **domasna2.pl** (со изворниот код за вашите предикати) и **domasna2-dokumentacija.pdf** (со документација за секоја задача и тоа: кои предикати ги имате дефинирано за да ја решите задачата и кратко објаснување за суштината на секој предикат – што работи и како го постигнува тоа). Двете датотеки спакувајте ги во единствена архива која ќе ја именувате со вашиот број на индекс **XXXXXX.zip** (каде XXXXXX е вашиот број на индекс) и таквата архива прикачете ја на соодветниот линк на страницата на курсот за предметот.

Плагијаторство е најстрого забрането! Истото ќе се проверува за секоја задача посебно. Решенија на задачи за кои ќе се утврди дека се плагијати по автоматизам ќе бидат вреднувани со 0 поени (без оглед на тоа кој е оригиналниот автор на решението). Дополнително секоја задача плагијат ќе ви го преполови вкупниот број на освоени поени (на пример: Од вашите оригинални решенија освоите 60 поени и имате една задача плагијат крајните поени кои ќе ги добиете ќе бидат 30, ако имате две задачи плагијат крајните поени кои ќе ги добиете ќе бидат 15, итн). Задачите ќе се вреднуваат според прогресот кон целосно решение (што значи не мора да ја решите целосно задачата за да добиете поени за истата). **За целосно решение се смета она кое би работело и на база на знаење поинаква (со различни вредности за фактите) од онаа во примерите.** Задачите за кои нема документација воопшто нема да се прегледуваат!