

Praktikum PR04

Selgitusi ja näiteid

Dünaamiline predikaat,

predikaadid, mida saab luua ja kustutada programmi töö kaigus.

Defineerimine:

`:- dynamic labitud/1.`

`assert/1` – lisab predikaatide dünaamilise andmebaasi lõppu uusi fakte ja reegleid.

`retract/1` - kustutab päringuga unifitseeruvaid predikaate ühekaupa.

`retractall/1` – kustutab dünaamilisest andmebaasist kõik antud päringuga unifitseeruvad predikaadid.

“is” - parema poole arvutamine (`Z is X+Y`)

Lõikepredikaat

Lõikepredikaat (!) elimineerib tagurdamisest (*backtracking*) kõik talle vastavas reeglis või päringus eelnevad predikaadid.

`p(X) :- a(X).`

`p(X) :- b(X),c(X),d(X),e(X).`

`p(X) :- f(X).`

`a(1).`

`b(1).`

`c(1).`

`b(2).`

`c(2).`

`d(2).`

`e(2).`

`f(3).`

`p(X) :- a(X).`

`p(X) :- b(X),c(X),!,d(X),e(X).`

`p(X) :- f(X).`

`a(1).`

`b(1).`

`c(1).`

`b(2).`

`c(2).`

`d(2).`

`e(2).`

`f(3).`

`?- p(X).`

`X = 1 ;`

`X = 2 ;`

`X = 3 ;`

`no`

`?- p(X).`

`X = 1 ;`

`no`

Rekursioon: NB! Vaata praktikumi **PR02** juhendit rekursiooni kohta.

Ülesanne

1. Luua teadmusbaas reisiinfo kohta, kus on kirjas, millise transpordivahendiga saab liikuda kahe linna vahel ja kui kaua selleks aega kulub (minutites).

Teadmusbaasi loomiseks kasuta predikaate:

- laevaga/3 - laevaga(Kust, Kuhu, Reisi_aeg_minutites).
- bussiga/3 - bussiga(Kust, Kuhu, Reisi_aeg_minutites).
- rongiga/3 - rongiga(Kust, Kuhu, Reisi_aeg_minutites).
- lennukiga/3 - lennukiga(Kust, Kuhu, Reisi_aeg_minutites).

Näiteks:

```
laevaga(tallinn, helsinki, 120).
laevaga(tallinn, stockholm, 480).
bussiga(tallinn, riia, 300).
rongiga(riia, berlin, 680).
lennukiga(tallinn, helsinki, 30).
lennukiga(helsinki, paris, 180).
lennukiga(paris, berlin, 120).
```

Lisa faktibaasi fakte nii, et tekiks võimalus linnade vahel ringiratast sõita.

Näiteks lisa: lennukiga(paris, tallinn, 120).

2. Lisa teadmusbaasile rekursiivne reegel reisi/2, mis leiab, kas on võimalik reisida ühest linnast teise. Ühenduse leidmisel võib olla vajalik kombineerida omavahel erinevaid transpordi vahendeid.

```
?- reisi(tallinn, berlin).
True.
```

```
?- reisi(tallinn, paris).
True.
```

3. Lisa teadmusbaasile reegel reisi/3, mis lisaks eelnevale näitab ka teel läbitavad linnad.

% reisi(Kust, Kuhu, mine(Kust, Kuhu)).

```
?- reisi(tallinn, paris, Tee).
Tee = mine(tallinn, helsinki, mine(helsinki, paris));
Tee = mine(tallinn, helsinki, mine(helsinki, paris)) ;
false.
?- reisi(tallinn, berlin, Tee).
Tee = mine(tallinn, helsinki, mine(helsinki, paris, mine(paris, berlin)));
Tee = mine(tallinn, helsinki, mine(helsinki, paris, mine(paris, berlin)));
Tee = mine(tallinn, riia, mine(riia, berlin));
false.
```

Lisa otsingureeglile kitsendus, et juba külastatud linna uuesti ei läbita.

Või kasutada nt. dünaamilist fakti labitud/1 teekonnal läbitud linnade hoidmiseks nt. Listis.

NB!! Näiteks reisi: tallinn → helsinki → paris → tallinn → helsinki → paris → berlin EI TOHI reegel leida.

4. Lisa teadmusbaasile reegel reisi_transpordiga/3, mis lisaks eelnevale näitab ka seda, millise transpordivahendiga antud vahemaa läbitakse.

```
?- reisi_transpordiga(tallinn, paris, Tee).
Tee = mine(tallinn, helsinki, laevaga, mine(helsinki, paris, lennukiga)) ;
```

```
Tee = mine(tallinn, helsinki, lennukiga, mine(helsinki, paris, lennukiga)) ;  
false.
```

5. Lisa teadmusbaasile reegel reisi/4, mis näitab läbitavaid linnu, millise transpordivahendiga antud vahemaa läbitakse ja reisiks kuluvat aega alguspunktist lõpppunkti.

```
?- reisi(tallinn, berlin, Tee, Hind).  
Tee = mine(tallinn, helsinki, lennukiga, mine(helsinki, paris, lennukiga,  
mine(paris, berlin, lennukiga))),  
Hind = 330 ;  
Tee = mine(tallinn, helsinki, laevaga, mine(helsinki, paris, lennukiga,  
mine(paris, berlin, lennukiga))),  
Hind = 420 ;  
Tee = mine(tallinn, riia, bussiga, mine(riia, berlin, rongiga)),  
Hind = 980 ;  
false.
```