

Praktikum 2

Eesmärk: Rekursiooni kasutamine sügavuti otsingul

Eelmises praktikumis tuli luua sugulaste teadmusbaas kasutades predikaate: mother/2, married/2, male/1, female/1 ja lisada reeglid sugulaste leidmiseks teadmusbaasist.

Näiteks:

```
% Seost, kes on kelle ema, esitasime faktiga: mother(Laps, Ema).
mother('Mai', 'Kai').
mother('Kai', 'Kadri').
mother('Kadri', 'Epp').
mother('Epp', 'Leida').
```

```
% vanaema leidsime reegiga: grandmother(Child, Grandmother).
grandmother(Child, Grandmother):-
    mother(Child, Mother),
    mother(Mother, Grandmother).
```

```
?- grandmother('Mai', 'Kadri').           % Kas Kadri on Mai vanaema?
true.
```

```
?- grandmother('Kai', Vanaema).           % Kes on Kai vanaema?
Vanaema = 'Epp'.
```

Kui me sooviksime teada vana-vanaema, siis tuleks kirjutada reegel:

```
grand_grandmother(C,G):-
    mother(C, Mother_1),
    mother(Mother_1, Mother_2),
    mother(Mother_2, G).
```

```
?- grand_grandmother('Mai', 'Epp').       % Kas Epp on Mai vana-vanaema?
True.
```

Rekursioon

```
esivanem(C, G) :- mother(C, G).
```

```
esivanem(C, G) :-
    mother(C, Mother),
    esivanem(Mother, G).
```

```
?- esivanem('Mai', 'Epp').               % Kas Epp on Mai esiema?
True.
```

```
?- esivanem('Mai', Vanem).               % Kes on Mai esiemad?
Vanem = kai ;
Vanem = kadri ;
Vanem = epp ;
Vanem = leida ;
false.
```

Rekursiooni kirjutamiseks on vaja vähemalt 2-te reeglit:

- baasreegel (reegel, mis peatab rekursiooni)
- ja teine, mis sisaldab rekursiooni.

vanem/2 reeglit rakendades saame teada, kas või kes on kellegi vanem.

Kui me sooviksime teada ka järgnevust e. kuidas on Leida Mai vana-vana-vanaema, siis saame kasutada järgnevat reeglit:

```
esivanem(X, Y, tee(X,Y)) :- mother(X, Y).
esivanem(X, Y, tee(X,Path)) :-
    mother(X, Z),
    esivanem(Z, Y, Path).
```

```
?- esivanem(mai, leida, Puu).
Puu = tee(mai, tee(kai, tee(kadri, tee(epp, leida)))) .
```

Täiendame emaks olemise fakti parameetriga, mis näitab kui vana oli ema lapse sünni ajal.

```
% mother(Laps, Ema, Ema_vanus).
mother(mai, kai, 24).
mother(kai, kadri, 35).
mother(kadri, epp, 27).
mother(epp, leida, 18).
```

Lisame reegli vanem/4, mis lisaks eelnevale arvutab, milline on vanusevahe esivanemaga.

```
esivanem(X, Y, tee(X,Y), V) :- mother(X, Y, V).
esivanem(X, Y, tee(X, Z, Path), V) :-
    mother(X, Z, Xv),
    esivanem(Z, Y, Path, Vz),
    V is Xv + Vz.
```

```
?- esivanem(mai, leida, Sugupuu, Vanus).
Sugupuu = tee(mai, kai, tee(kai, kadri, tee(kadri, epp, tee(epp, leida)))),
Vanus = 104 .
```

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).
```

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.
```

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, paris, Tee, Aeg).
Tee = mine(tallinn, helsinki, laevaga, mine(helsinki, paris, lennukiga)),
Aeg = 300 ;
Tee = mine(tallinn, helsinki, lennukiga, mine(helsinki, paris, lennukiga)),
Aeg = 210 ;
false.
```