

Harjutus 2. kontrolltöök

ITI0211

J.Vain

24.11.2020

DCG (loeng 8)

- Terminal- ja mitteterminalsümbolite mõiste
- DCG reeglite esitus ja kitsendused reeglite kirjutamisel
- DCG reeglite esitamine Prolog programmina

Tagurdamise kasutamine reeglites

- Otsingu juhtimise predikaadid *repeat*, *fail*, *cut* ja nende kasutamine
- Tagurdamisega otsing faktide hulgal
- Ülesanne (5 punkti):
 - Olgu antud hulk h faktidega hulk/2, kus esimene parameter on hulga nimi ja teine parameter on selle hulga elemendi väärtus.

```
:- dynamic hulk/2.  
hulk(h,1).  
hulk(h,2).  
hulk(h,3).  
hulk(h,4).
```

Programmeerige tagurdamisega reegel `delete_smaller(H, Const)`, mis kustutab kõik need hulga H elemendid, mis on väiksemad selles päringus etteantud konstandist `Const`.

Näide:

```
?- delete_smaller(h, 2).  
kustutab fakti hulk(h,1).
```

Tagurdamise kasutamine reeglites: ülesande lahendused

• Lahendus 1:

```
delete_smaller(H, Const) :-  
    repeat,  
    retractall(tunnus),  
    hulk(H, E1),  
        (E1 < Const,  
        retract(hulk(H, E1)),  
        assert(tunnus)  
        ;  
        true),  
    not(tunnus).
```

Lahendus 2:

```
delete_smaller(H, Const) :-  
    hulk(H, E1),  
    E1 < Const,  
    retract(hulk(H, E1)),  
    fail.  
delete_smaller(_, _).
```

Rekursioon listidel

- Ülesanne (5 punkti) :
 - Koostada programm, mis tuvastab kas list L2 on listi L12 sufiks (lõpuosa) ja tagastab listi L1, kus L1 on listi L12 algusosa (prefiks)
 - Päring:

```
?- suffix(L1, L2, L12) .
```
 - Märkus: Lubatud on kasutada süsteempredikaati `reverse`

Rekursioon listidel: ülesanne

- Ülesanne (5 punkti):
 - Olgu antud listide kujul hulgad H1 ja H2. Koostada programm, mis leiab hulkade H1 ja H2 vahe ja tagastab selle listis H3.
(Hulk H3 on hulkade H1 ja H2 vahe, kui selles sisalduvad ainult need hulga H1 elemendid, mis ei sisaldu hulgas H2)
 - Päring:
?- vahe (H1, H2, H3).

Hierarhilised listid: ülesanne

- Ülesanne:
 - Koostada predikaat `list_partition/2`, mis jagab tasapinnalise listi erladussümboli `'\n'` alusel alamlistide listiks .
 - Päring: `list_partition(ListIn, ListOut)`
 - Näide:

```
?- list_partition([s,d,'\n',d,f,'\n','See'],ListOut).  
ListOut = [[s,d],[d,f],['See']]
```
- Soovitus: kasutada reeglis sabarekursiooni

Hierarhilised listid: lahendus ülesandele

```
list_partition([], [[]]).
```

```
list_partition(['\n' | ListIn], [[] | ListOut]) :-
```

```
    list_partition(ListIn, ListOut).
```

```
list_partition([Element | ListIn], [[Element | InternalL] | ListOut]) :-
```

```
    list_partition(ListIn, [InternalL | ListOut]).
```