ITI0211 Loogiline programmeerimine:

Harjutus Testiks 1

J.Vain

Sügis 2020

Loogilise programmeerimise keel

- Leia termi $f(g(X), g(a), gh(Z, p(U)), f(g(Y)))$ kõik alamtermid ja nendes sisalduvad muutujad.
- Millest koosneb Horni lause?
- Mille poolest erinevad positiivne ja negatiivne literaal?
- Mis loogikatehteid saab kasutada Horni lause kehas?
- Kuidas defineeritakse dünaamiline predikaat?

Unifitseerimine

- Konstrueerige termidele $P(a, X)$ ja $P(Y, Q(\ Y, b))$ mgu-algoritmiga kõige üldisem unifitseerija.
- Kas termid Term1 ja Term2 unifitseeruvad? Kui jah, siis leidke mgu-algoritmiga unifitseerija
  - Term1: `relative(peter,U,onu(peter, felix))`
  - Term2: `relative(Y,isa(felix),onu(Y,U))`
- Kumb unifitseerija on üldisem, kas
  - $\{Z/d, G/ff1(d(q,w,U)), U/V\}$

  või
  - $\{G/ff1(d(q,w,U)), U/4\}$?

Resolutsioon

- Kas Horni lausetele P(*a*, *X*, *Y*)$\vee$S(*X*,*Y*) ja ¬P(*U*, *V*, *b*) $\vee$ R(*U*,*V*) saab rakendada resolutsiooni? Kui jah, siis leidke see lause, mille saab tuletada antud lausetest.

- Missugused järeldustest a) – c) on tuletatavad allpool toodud teadmusbaasis?

```
a)  mother(jack, linda).
b)  mother(peter, mary).
c)  brother(bob, alice).
```

```
mother(mary, ann).
mother(alice, mary).
mother(carol, mary).
mother(linda, ann).
mother(fred, linda).
mother(bob, mary).
```

```
brother(peter,carol).
brother(jack, fred).
brother(james, ann).
```

```
mother(A, B):-
    brother(A, C),
mother(C, B).
```

Prolog keel

- Missugune alltoodud vastustest a) – e) tagastatakse järgnevale päringule (kriipsuta alla õige)?

```
?- D is 4 // 3, E is 4 / 3, D =:= E.
```

a) true
b) false
c) ERROR: Execution Aborted
d) Stack full
e) Type error

Spikker:

// - täisarvuline jagamine
/ - ratsionaalarvuline jagamine
=:= - arvuline võrdus

Rekursioon

Kriipsuta alla kõik õiged vastuse variandid. Järgnev programm

```
transitive_closure(Rel):-
                Relation =..[Rel,X,Y],
                Relation,
                assertz(closure(1,X,Y)),
                fail.
transitive_closure(_):-
                closure(N,A,B)),
                closure(1,B,C)),
                N1 is N+1,
                assertz(closure(N1,A,C)),
                fail.
```

a) töötab sabarekursiooniga

b) töötab tagurdamisega

c) tagastab “false”

d) tagastab “true“

e) genereerib faktid closure/2

e) genereerib faktid nimega Relation

Rekursiivsed reeglid

- Koostada sabarekursiivne programm, mis arvutab *n*-faktoriaali.

Päring:

```
?- factorial(N, NFactorial).
```

- Koostada sabarekursiivne programm, mis arvutab argumendi `X` *n*-da astme ja tagastab selle parameetris `Vastus`

Päring:

```
?- astendamine(X,N, Vastus).
```

Rekursioon listidel

- Koosta rekursiivne reegel `suffix(List1, List2)`, mis tagastab *true*, kui `List1` on `List2` sufiksiks ja muul juhul tagastab *false*.

    - Vihje: Vastuse õigsust saab kontrollida päringuga

    ```
    ?- append(_,L1,L2).
    ```

- Koostada rekursiivne programm, mis võrdleb kahe listi `List1` ja `List2` pikkust ja tagastab *true*, kui `List2` on pikem kui `List1`.

    Päring:

    ```
    ?- suurem(List1, List2).
    ```