Loeng 3:
Teema 4: Loogiline programmeerimine
4.1. Horni lause

Loogiline programm on Horni disjunktsioonide (disjunktiivsete valemite) kogu, kus ükski valem ei sisalda üle ühe positiivse literaali.

Literaal – atomaarne valem või selle eitus (atomaarne valem: $P(t_1,\ldots, t_n)$)

Horni reegel (*lause*): $p \leftarrow q_1,\ldots, q_n$

$$\equiv \neg(q_1, \ldots, q_n) \vee p$$
$$\equiv \neg(q_1 \wedge \ldots \wedge q_n) \vee p$$
$$\equiv \neg\, q_1 \vee \ldots \vee \neg q_n \vee p$$

- HL protseduurne interpretatsioon:

”Programmi p arvutamiseks tuleb arvutada tema alamprogrammid q_1, ..., q_n ”

- HL interpretatsioon loogikas:

  ”Kui q_1,..., q_n on tõesed, siis on ka p tõene” ehk
  ”Lausetest q_1,..., q_n järeldub lause p”
  "Lause p kehtib siis, kui kehtivad laused q_1,..., q_n "

HL struktuur:

$$p \leftarrow q_1,\dots, q_n$$

p – Horni lause päis
q_1,..., q_n – Horni lause keha

- Horni lausete tüübid:
  - *Fakt* – HL, millel puudub keha e. ainsast positiivsest literaalist koosnev lause:

    Näide: $Q(a, b) \leftarrow$ ehk $Q(a, b)$ % tingimusteta tõene lause (PA)

    isa(jaan, peeter). (Prolog)

  - *Reegel* - HL, millel on pea ja keha e. ühest positiivsest ja vähemalt ühest negatiivsest literaalist koosnev disjunktsioon.

    Näide: $Q(a, b) \leftarrow R(a, v), \ldots, P(b, b, w)$ % tingimuslause (PA)

    vanaisa(X, Y):- isa(X, Z), ... ,ema(Z, Y). (Prolog)

    Muutujate esinemine reeglis:

    $\forall x_1 \ldots x_n \exists y_1 \ldots y_m\, z_1 \ldots z_l : Q_1(y_1 \ldots y_m) \wedge \ldots \wedge Q_n(z_1 \ldots z_l) \Rightarrow P(x_1 \ldots x_n)$ (PA)

    $p(X1\ldots Xn)$:- $q1(Y1\ldots Ym)$, ... , $qn(Z1\ldots Zl)$ (Prolog)

  - *Päring* – HL, millel on ainult keha e. vähemalt ühest negatiivsest literaalist koosnev disjunktsioon.
    - päring Prologis on otsingut käivitav käsk, näiteks ?- isa(juku,X).

Suletud maailma eeldus:
tõene on ainult see väide, mille tõesuse saab Prolog programmis tuletada olemasolevatest faktidest.

NB! Faktide puudumine tähendab faktide mittekehtimist so
faktide puudumine ei tähenda määramatust!

4.2 Resolutsiooni meetod

Resolutsioon - Horni lausete kujul oleva deduktiivse süsteemi tuletusreegel.

- Valem $\Delta \Rightarrow D$ kehtib parajasti siis, kui tema eitus $\neg(\Delta \Rightarrow D) \equiv \Delta \wedge \neg D$ on vasturääkiv.
- Horni lause tõestamiseks tõestatakse, et positiivse literaali eituse konjunktsioonist eeldusdisjunktidega saab tuletada **vastuolu**.
- st temast saab tuletada **tühja disjunkti**.

Tühja disjunkti tuletamiseks kasutame klassikalise loogika reegli *modus ponens* üldistust - *resolutsiooni reeglit* (RR).

Modus ponens:

$$\frac{A \qquad A \Rightarrow B}{B}$$

RR (Horni lausete kujul):

$$\frac{A_1 \leftarrow A_2, \ldots, A_n \qquad \leftarrow A_1, B_2, \ldots, B_m}{\leftarrow A_2, \ldots, A_n, B_2, \ldots, B_m}$$

Asendame valemis osalause tema eelduslausetega

Reegel (disjunktide kujul):

$$\frac{\neg A_1 \vee A_2 \vee \ldots \vee A_n \qquad A_1 \vee B_2 \vee \ldots \vee B_m}{A_2 \vee \ldots \vee A_n \ \vee \ B_2 \vee \ldots \vee B_m} \qquad \text{(RR)}$$

Pärast RR iga rakendamist on saadud valemis 2 literaali vähem kui RR eeldusvalemites.

Faktoriseerimisreegel:
Kui disjunktsioonis on kaks ühesugust literaali, võib neist ühe eemaldada

$$\frac{A_1 \vee A_1 \vee A_2 \vee \ldots \vee A_n}{A_1 \vee A_2 \vee \ldots \vee A_n}$$

Näide:

Olgu antud faktide hulk:

1) isa(jaan, martin)
2) $\neg$isa(riivo, veiko)
3) isa(martin, veiko) $\vee$ isa(riivo, veiko)
4) $\neg$isa(martin, veiko) $\vee$ vanaisa(jaan,veiko)

Kas faktidest 1) - 4) saab tuletada fakti vanaisa(jaan,veiko)?
St kas disjunktide hulk 1) – 4) koos faktiga

5) $\neg$vanaisa(jaan,veiko)

on vastuoluline ehk kas neist saab tuletada tühja disjunkti?

2 ja 3 $\Rightarrow$ 6) isa(martin, veiko)
4 ja 6 $\Rightarrow$ 7) vanaisa(jaan,veiko)
7 ja 5 $\Rightarrow$ tühidisjunkt ($\rightarrow$)

4.3 Resolutsioon muutujaid sisaldavate disjunktidega

Näide:

Olgu valemite hulk $\boldsymbol{\Gamma}$:

1. isa(Jaan,Peeter).
2. isa(Jaan, Martin).
3. isa(Martin,Veiko).
4. isa(Riivo, Leo).
5. ema(Leena, Leo) $\vee$ isa(Leena, Leo).
6. $\neg$isa(Leena, Leo).
7. $\neg$isa(X, Y) $\vee$ $\neg$isa(Y, Z) $\vee$ vanaisa(X, Z).
8. $\neg$isa(X, Y) $\vee$ $\neg$ema(Y, Z) $\vee$ vanaisa(X, Z).

ja valem

$\boldsymbol{A}$: vanaisa(Jaan, Veiko)

Tõestame $\boldsymbol{\Gamma} \Rightarrow \boldsymbol{A}$:

Meenutame, et

$\Gamma \Rightarrow A$ kehtib parajasti siis, kui $\Gamma \wedge \neg A$ on vastuoluline.

Laiendame hulka Γ lisades $\neg A$ st. $\Gamma' = \Gamma, \neg A$:

9. ¬vanaisa(Jaan, Veiko)

Püüame Γ'-st tuletada vastuolu e. tühja disjunkti.

- Kuidas kasutada tühja disjunkti tuletuses muutujaid sisaldavaid reegleid?

Muutujat *V* sisaldav väide tähistab *kõikide konkretiseeritud väidete* hulka, kus konkretiseerimise all mõeldakse *V* asendamist kas konstantide või muutujaid sisaldavate termidega.

Näide:

¬isa(X, Y) ∨ ¬isa(Y, Z) ∨ vanaisa(X, Z) konkretiseeritud väited on

1) ¬isa(Leo, Jaan) ∨ ¬isa(Jaan, Martin) ∨ vanaisa(Leo, Martin),
2) ¬isa(Leo, Peeter) ∨ ¬isa(Peeter, Martin) ∨ vanaisa(Leo, Martin)

jne.

- Kuidas leida konkretiseering, mida kasutades saab rakendada resolutsiooni?

Olgu reeglis

$$\frac{\neg A_1 \vee A_2 \vee \ldots \vee A_n \qquad A'_1 \vee B_2 \vee \ldots \vee B_m}{A_2 \vee \ldots \vee A_n \ \vee \ B_2 \vee \ldots \vee B_m}$$

literaalid A_1 ja A_1', mis erinevad ainult konkretiseeringu või muutujate nimede poolest.

Et rakendada resolutsiooni, peab saama literaale A_1 ja A_1' konkretiseerida ühesuguseks st *unifitseerida.*

unifitseerimine

Näide (järg - unifitseerimine):
Et rakendada resolutsioonireeglit väidetele 2 ja 7:

2. isa(Jaan, Martin)
7. ¬isa(X, Y) ∨ ¬isa(Y, Z) ∨ vanaisa(X, Z),

tuleb nende literaalid unifitseerida. Unifitseerimine toimub muutujate asendamise teel.

Muutujate asendus e. substitutsioon: $\{x_1/t_1, \dots , x_n/t_n\}$, kus
x_i – asendatav muutuja
t_i – asendav objekt, kus t_i on erinev x_i –st

Selgitus: Kasutame asendust {X/Jaan,Y/Martin}

2. on konkreetne, st ei muutu 2' = 2
7.' ¬isa(Jaan, Martin) ∨ ¬isa(Martin, Z) ∨ vanaisa(Jaan, Z).

Definitsioon (Termide unifitseeruvus):

Termid t_1 ja t_2 on unifitseeruvad, kui leidub asendus σ, nii et $\sigma(t_1) = \sigma(t_2)$

Näites: {X/Jaan, Y/Martin} ← *unifitseerija* (teatud asenduste hulk)

Peale RR reegli rakendamist, saame

7''. ¬isa(Martin, Z) ∨ vanaisa(Jaan, Z).

Näide (järg):

Unifitseerime 3 ja 7'':

3. isa(Martin,Veiko).

7''. ¬isa(Martin, Z) ∨ vanaisa(Jaan, Z).

Unifitseerija {Z/Veiko} annab

7'''. ¬isa(Martin, Veiko) ∨ vanaisa(Jaan, Veiko)

ning RR reegli abil tuletame väidetest 3. ja 7'''.

10. vanaisa(Jaan, Veiko)

9. ja 10. annavad RR põhjal tühja disjunkti e. vastuolu.

Näite lõpp

- Kuidas unifitseerida, kui mõlemad literaalid sisaldavad muutujaid?

Minimaalse substitutsiooni reegel:
Asendamata jäetakse kõik muutujad, mille jaoks puudub konkretiseeriv asendus.

Näide:

Unifitseerime	$P(a, X, Y) \vee S(X,Y)$ ja $\neg P(U, V, b) \vee R(U,V)$
kasutades asendust	$\{U/a, Y/b, X/V\}$.
Saame	$P(a, V, b) \vee S(V, b)$ ja $\neg P(a, V, b) \vee R(a,V)$
RR rakendamine annab	$S(V, b) \vee R(a,V)$

Definitsioon (Kõige üldisem unifitseerija - mgu):

Termide t_1 ja t_2 kõige üldisem unifitseerija on asendus ρ, mis rahuldab tingimusi:

1. ρ on termide t_1 ja t_2 unifitseerija
2. t_1 ja t_2 iga unifitseerija σ korral leidub veel asendus τ, nii et $\sigma = \tau \circ \rho$
   st. iga termi t korral $\sigma(t) = \tau(\rho(t))$

Märkus: mgu annab kõige vähem konkretiseeritud omavahel unifitseeruvad termid.

Lause:

Leidub algoritm mgu, mis arvutab literaalide või termide x ja y kõige üldisema unifitseerija.

Näiteid:

mgu(P(a,X), P(Y,b)) = $\{Y/a,\ X/b\}$

mgu(P(X, f(X)), P(f(Y), U)) = $\{X/f(Y),\ U/f(f(Y))\}$

mgu(L(g(X,X)), L(g(f(a), f(a)))) = $\{X/f(a)\}$

mgu(R(a,b), R(a,b)) = $\{\}$

Resolutsioonireegli üldkuju:

$$\frac{\neg A_1 \vee A_2 \vee \ldots \vee A_n \qquad B_1 \vee B_2 \vee \ldots \vee B_m}{(A_2 \vee \ldots \vee A_n \;\vee\; B_2 \vee \ldots \vee B_m)\sigma} \quad \sigma = \mathrm{mgu}(A_1, B_1)$$

Reegel: Disjunktsioonides, millele rakendatakse üldistatud RR reeglit, tuleb eelnevalt asendada samanimelised muutujad.

Unifitseerimisalgoritm mgu

```
substitution mgu(x,y)
term x,y;                          % x ja y on unifitseerivad termid
{
  int i;
  substitution g;

% Kui termideks on muutujad, konstandid või termid on erineva pikkusega
if x==y return EMPTYSUBST;
else if variable(x) return mgu var(x,y);     % Kas x tüüpi muutuja
else if variable(y) return mgu var(y,x);     % Kas y tüüpi muutuja
else if (constant(x)||constant(y)) return NOSUBST;
                                             % Erinevad konstandid
else if (length(x) != length(y)) return NOSUBST;
                                             % Ei ole unifitseeritavad
```

% ühepikkuste mitmekohaliste termide unifitseerimine

```
i=0;
g=EMPTYSUBST;
    while (i<=length(x))  {    % analüüsime alamtermide kaupa
    s= mgu(part(x, i), part(y,i));
    if s==NOSUBST return s;    % Kui puudub asendus
    else                       % Kui leidub asendus
        g=compose(g,s);        % Täiendada unifitseerijat g asendusega s
        x=substitute(x,g);     % Rakendada unifitseerijat g termidele x ja y
        y=substitute(y,g);
    i= i+1;
}
return g;
}
```

```
substitution mgu_var(x,y)
term x,y;
{
   if occurs_in(x,y) return NOSUBST;
                                        % Muutuja ei tohi esineda temaga
   else return makesubstitution(x,y);
                                        % unifitseeritavas termis, sest siis
                                        % tekib "ringasendus"
}
```

Näide:

mgu(P(a,X,X), P(Y,b,Y)) ei eksisteeri

Harjutus

Missugused alltoodud asendustest on korrektsed

- $\{x/y,\ y/x\}$
- $\{x/x,\ y/x\}$
- $\{x/y,\ x/z\}$
- $\{x/f(x,y),\ y/\mathrm{g}(z)\}$
- $\{x/a,\ y/x\}$
- $\{z/a,\ b/c\}$

Rakendada asendust $\{x/h(y,\ a),\ y/b,\ z/g(c)\}$ termidele

- $f(x,y,z)$
- $f(a,b,c)$
- $h(x,x)$
- $f(g(h(x),v),\ g(x,y),\ g(z,\ f(x,y,u)))$

Leida mgu

mgu($p(f(y),w, g(z)),\ p(z,h(z,w), f(v))$) = ?

mgu($p(a, x, f(g(y)), p(u, u, v)$)) = ?