Loeng 3: Loogilise programmeerimise alused

J.Vain

Loengu kava

- Loogika valemid Horni lause kujul
  - Fakt
  - Reegel
  - Päring
- Horni lausete tõestmine resolutsiooni meetodiga
- Termide unifitseerimine kui resolutsiooni eeldus
- Näided
- *Takeaway*
  - Teadmised Prologi tõestussmehhanismist

Horni lause

- *Loogiline programm* koosneb *Horni lausetest*
- Horni lause üldkujul:

$p \leftarrow q_1,\ldots, q_n,$

kus

p – Horni *lause päis*

$q_1,\ldots, q_n$ – Horni *lause keha*

Horni lause

- Horni lause $p \leftarrow q_1,\ldots, q_n$ esituskujud:
  - *implikatiivne* kuju:

$$p \leftarrow q_1 \wedge \ldots \wedge q_n$$

    - kus p ja $q_1,\ldots, q_n$ tähistavad *literaale*
  - *disjunktiivne* kuju (sisaldab $\leq$1 positiivse literaali):

$$p \vee \neg\, q_1 \vee \ldots \vee \neg\, q_n$$

- Implikatiivne ja disjunktiivne kuju on *semantiliselt ekvivalentsed*

- *Literaal* – atomaarne valem või selle eitus
- *Atomaarne valem*: $P(t_1,\ldots, t_n)$, kus P – *predikaadi sümbol*
  $t_1, \ldots, t_n$ – termid

Kuidas mõista Horni lauset $p \leftarrow q_1,..., q_n$?

- Operatsiooniline semantika:

  "Programmi p täitmiseks tuleb täita alamprogrammid $q_1, ..., q_n$"

- Deklaratiivne semantika:

  "Lause p tõesus järeldub lausete $q_1,..., q_n$ tõesusest" ehk

  "*Lause p kehtib siis, kui kehtivad laused $q_1,..., q_n$*"

- Muutujate (vaikimisi) tõlgendamine reeglis:

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

  $\forall x_1 ... x_n \exists y_1 ... y_m z_1 ... z_l : P(x_1... x_n) \leftarrow Q_1(y_1 ... y_m) \wedge ... \wedge Q_n(z_1 ... z_l)$ (PA)

  kõik muutujad Horni lauses on kvantoritega seotud

Horni lause tüübid

- *Fakt*

  lause, millel puudub keha (disjunktiivse kuju puhul ainsast positiivsest literaalist koosnev disjunktsioon).

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

  `isa(jaan, peeter).` (Prolog)

- *Reegel*

  lause, millel on pea ja keha (disjunktiivse kuju puhul 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)

Horni lause tüübid

- *Päring* – Horni lause, millel on ainult keha (disjunktiivse kuju puhul vähemalt ühest negatiivsest literaalist koosnev disjunktsioon).

- Prologi päring käivitab lahendi otsingu

  Näide: `?- isa(juku,X).`

Suletud maailma eeldus

<u>Suletud maailma eeldus</u>:

Tõene on ainult see väide, mille tõesuse saab loogiline programm tuletada teadmusbaasis olevatest faktidest.

- NB!
  - Faktide puudumine teadmusbaasist <u>ei tähenda</u> määramatust
  - Fakti puudumine teadmusbaasist tähendab fakti eitust!

Resolutsioon

- *Resolutsioon* - Horni lausete kujul oleva deduktiivse süsteemi tõestusreegel.
- Kuidas mõista resolutsiooni?
  - Vastuväiteline tõestamine:

  Olgu $D \leftarrow \Delta$ Horni lause, kus *D* on reegli päis ja Δ on reegli keha.

  Valem $D \leftarrow \Delta$ kehtib parajasti siis, kui tema eitus $\neg(D \leftarrow \Delta)$ on väär.

  - Horni lause tõestamiseks tõestatakse, et valemist $\neg(D \leftarrow \Delta)$ saab tuletada ***vastuolu***.
  - Vastuolu Prologi mõttes tähendab, et teadmusbaasis ei leidu niisugusi fakte, millest saaks tuletada tõestatava väite.
  - *resolutsiooni reegel* (RR) võimaldab tuletada tõestatava valemi eitusest *tühja valemi* kasutades vahe-eelduste „väljalõikamist“

Resolutsiooni reegel

Olgu Horni lause (HL) implikatsiooni kujul

- HL päring: $\leftarrow A_1, B_2, \ldots, B_m,$ milles esineb literaal A_1
- HL reegel: $A_1 \leftarrow A_2, \ldots, A_n,$ kus literaal A_1 on järelduses,

siis resolutsiooni reeglit

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

rakendades asendame päringus literaali A_1 reegli $A_1 \leftarrow A_2, \ldots, A_n$ põhjal eeldustega $A_2, \ldots, A_n$.

Tulemusena omandab algne päring $\leftarrow A_1, B_2, \ldots, B_m$ kuju:

$$\leftarrow A_2, \ldots, A_n, B_2, \ldots, B_m$$

Kui päringu kõigi literaalide asendamisel on jõutud faktideni, siis neil puuduvad eelduslaused ja faktide asendamine annab tühja literaali.

Resolutsiooni reegel

Resolutsiooni reegel disjunktsiooni kujul olevate Horni lausetega

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

Olgu $\Gamma \equiv \neg A_1 \vee A_2 \vee \ldots \vee A_n$ ja $\Lambda \equiv A_1 \vee B_2 \vee \ldots \vee B_m$

Kehtib $\Gamma \wedge \Lambda \Rightarrow \Gamma \vee \Lambda$.

Kuna $\Gamma \vee \Lambda$ sisaldab $\neg A_1 \vee A_1$ ja

$\neg A_1 \vee A_1 \Leftrightarrow \mathit{true}$ ning $\mathit{true} \vee A \Leftrightarrow A$

saame $A_2 \vee \ldots \vee A_n \;\vee\; B_2 \vee \ldots \vee B_m$

Faktoriseerimisreegel (abireegel)

- Kui päringus on kaks ühesugust literaali, võib neist ühe eemaldada ilma valemi tõeväärtust muutmata.

$$\frac{\leftarrow A_1, A_1, A_2, \ldots, A_n}{\leftarrow A_1, A_2, \ldots, A_n}$$

Resolutsiooni rakendamine

Näide:

Olgu antud Horni laused disjunktsiooni kujul:

1. `isa(jaan, martin)`
2. `¬isa(riivo, veiko)`
3. `isa(martin, veiko)` ∨ `isa(riivo, veiko)`
4. `¬ isa(martin, veiko)` ∨ `vanaisa(jaan,veiko)`

Kas lausetest 1) - 4) saab tuletada lause `vanaisa(jaan,veiko)`?

st kas laused 1) – 4) koos negatiivse literaaliga

5. `¬ vanaisa(jaan,veiko)`

on vastuolulised ehk kas neist saab tuletada tühja disjunkti?

2 ja 3 $\xrightarrow{\text{resolutsioon}}$ 6. `isa(martin, veiko)`

4 ja 6 $\xrightarrow{\text{resolutsioon}}$ 7. `vanaisa(jaan,veiko)`

7 ja 5 $\xrightarrow{\text{resolutsioon}}$ tühi disjunkt

Resolutsioon muutujaid sisaldavate Horni lausetega

Näide

Olgu lausete hulk Γ:

isa(jaan, peeter).

isa(jaan, martin).

isa(martin, veiko).

isa(riivo, leo).

ema(leena, leo) $\vee$ isa(leena, leo).

$\neg$ isa(leena, leo).

$\neg$ isa(X, Y) $\vee$ $\neg$ isa(Y, Z) $\vee$ vanaisa(X, Z).

$\neg$ isa(X, Y) $\vee$ $\neg$ ema(Y, Z) $\vee$ vanaisa(X, Z).

ja valem ***A***, mida tahame tõestada hulgal Γ, kus

***A*:** vanaisa(jaan, veiko)

Seega näitame, et kehtib $\Gamma \rightarrow$ ***A*:**

Resolutsioon muutujaid sisaldavate Horni lausetega

- Eelnevast teame, et kehtib samasus $\Gamma \rightarrow A \equiv \neg\, \Gamma \vee A$,
- seega valem $\Gamma \rightarrow A$ kehtib parajasti siis, kui tema eitus $\neg(\Gamma \rightarrow A) \equiv$

  $\equiv \neg\,(\neg\,\Gamma \vee A) \equiv \Gamma \wedge \neg\, A$ on vasturääkiv.
- Vasturääkivuse näitamiseks laiendame hulka Γ lisades sellele valemi $\neg\, A$, saame lausete hulga $\Gamma' = \Gamma \cup \neg\, A$:

  Näite puhul: $\Gamma' = \Gamma \cup \neg\,$ vanaisa(jaan, veiko)
- Tuletame resolutsiooni kasutades Γ'-st vastuolu e. tühja disjunkti.

- Kuidas aga rakendada resolutsiooni *muutujaid sisaldavate* Horni lausete puhul?

Resolutsioon muutujaid sisaldavate Horni lausetega

- Muutujat V sisaldav lause tähistab *kõigi* lausete hulka, milles V on asendatud konkretiseeriva termiga.

Näide (muutujate X, Y, Z konkretiseerimine):

Lause $\neg$ `isa(X, Y)` $\vee$ $\neg$ `isa(Y, Z)` $\vee$ `vanaisa(X, Z).`

konkretiseeringud on:

1) $\neg$`isa(leo,jaan)` $\vee$ $\neg$`isa(jaan,martin)` $\vee$ `vanaisa(leo, martin).`

2) $\neg$`isa(leo,peeter)` $\vee$ $\neg$`isa(peeter,martin)` $\vee$ `vanaisa(leo,martin).`

kui X$\in${leo}, Y$\in${jaan, peeter}, Y$\in${martin}

Resolutsioon muutujaid sisaldavate Horni lausetega

- 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 nendes esinevate muutujate nimede poolest.
- Tuletame meelde muutujate interpretatsiooni reeglis:

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

- Et rakendada resolutsiooni, peab muutujad literalides A_1 ja A_1' konkretiseerima nii, et literalid oleksid ühesugusel kujul ehk *unifitseeritud*.

Resolutsioon muutujaid sisaldavate Horni lausetega

<u>Näide (järg - unifitseerimine)</u>:

Et rakendada resolutsioonireeglit lausetele 2 ja 7, kus

2. `isa(jaan, martin)`
7. `¬ isa(X, Y) ∨ ¬ isa(Y, Z) ∨ vanaisa(X, Z),`

tuleb nende literaalid unifitseerida. Unifitseerimine toimub *muutujate asendamise* teel termidega.

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

<u>Selgitus</u>: Näites kasutame asendust $\{X/jaan, Y/martin\}$

2. on juba konkreetne, sest konstante ei saa asendada,

7 $\xmapsto{\text{unifitseerimine}}$ 7' : `¬isa(jaan,martin)∨ ¬isa(martin,Z)∨ vanaisa(jaan,Z).`

Termide unifitseerimine

__Definitsioon__ (Termide unifitseeruvus):

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

Termide unifitseerimine

Näide (järg): kasutame unifitseerijat {X/jaan, Y/martin}

Peale X ja Y unifitseerimist ja RR reegli rakendamist, saame

7''. ¬ isa(martin, Z) ∨ vanaisa(jaan, Z).

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'''.

9. ¬ vanaisa(jaan, veiko)

10. vanaisa(jaan, veiko)

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

Termide unifitseerimine

- Kuidas unifitseerida, kui unifitseeritavad termid sisaldavad muutujaid?

Minimaalse substitutsiooni reegel:

Asendatakse ainult need muutujad, mille asendamine võimaldab rakendada resolutsiooni reeglit.

<u>Näide</u>:

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	$S(V, b) \vee R(a,V)$

Kõige üldisem unifitseerija - mgu

NB! Asendused tehakse asendajatega vasakult paremale järjestuses ja asendus tehakse teadmusbaasi kõikides Horni lausetes.

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 teise unifitseerija σ korral leidub veel asendus τ, nii et $\sigma = \tau \bullet \rho$ st. iga termi t korral $\sigma(t) = \tau(\rho(t))$.

($\bullet$ – tähistab unifitseerijate kompositsiooni)

Selgitus:

! mgu annab kõige vähem konkretiseeritud omavahel unifitseeruvad termid

Unifitseerimise 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)) = {}

Resolutsioon unifitseeritavatel literalidel

Unifitseerimisega resolutsiooni reegel:

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

Horni lausetes, millele rakendatakse resolutsiooni reeglit, tuleb eelnevalt unifitseerida mgu-ga kõik termid

Kõige üldisem unifitseerija - mgu

<u>Lause</u>:

Leidub algoritm mgu, mis arvutab literalide ja termide *x* ja *y* kõige üldisema unifitseerija.

- Lause tõestuseks esitame järgnevalt mgu-algoritmi

Algoritm mgu leidmiseks (1)

```
substitution mgu(x,y)
term x, y;                          % Olgu x ja y unifitseeritavad termid
{ int i;
 Substitutions list g;              % Asendused salvestame listi g
```

% Case1: termideks on muutujad, konstandid või termid on erineva pikkusega

```
if x==y return EMPTYSUBST;                          %  kui X ja Y on ühesugused
else if variable(x) return mgu_var(x,y);  % Kui x on muutuja
else if variable(y) return mgu_var(y,x);  % Kui y on muutuja
else if (constant(x)||constant(y)) return NOSUBST;
                                                    % Kui erinevad konstandid
else if (length(x) != length(y))                    % Kui aarsus erinev,
return NOSUBST;                                     % siis ei ole unifitseeritavad
```

Algoritm mgu leidmiseks (2)

```
% Case 2: ühepikkuste mitmekohaliste termide unifitseerimine
i=0;
g=EMPTYSUBST;
while (i<=length(x))                        % iteratsioon alamtermide kaupa
{
   s= mgu(subterm(x,i), subterm(y,i)); % X ja Y i-nda alamtermi võrdlus
   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 termile x
       y=substitute(y,g);                   % Rakendada unifitseerijat g termile y
i= i+1;
}
return g;
}
```

Case 3: Muutujate asendatavuse kontroll

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

Näide:

$\mathrm{mgu}(\mathrm{P}(a,X,X), \mathrm{P}(Y,b,Y))$ - siin ei eksisteeri unifitseerivat asendust

Unifitseerimise harjutusi

Missugused alltoodud asendustest on lubatud?

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

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

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

Unifitseerimise harjutusi

Leida mgu termidele:

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

$$\text{mgu}(p(a,\ x,\ f(g(y))),\ p(u,\ u,\ v)) = ?$$