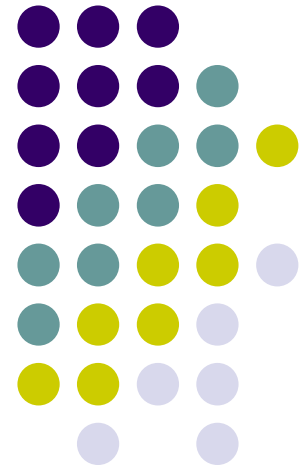
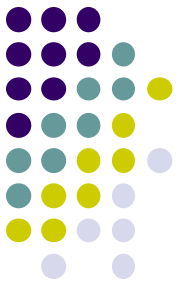


# Loeng 3: Loogilise programmeerimise alused



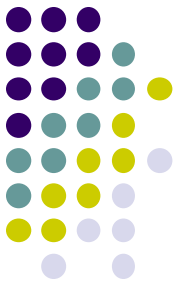
J.Vain





# Loengu kava

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



# Horni lause

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

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

$p$  – Horni *lause päis*

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



# Horni lause

- Horni lause  $p \leftarrow q_1, \dots, q_n$  esituskujud:

- *implikatiivne* kuju:

$$p \leftarrow q_1 \wedge \dots \wedge q_n$$

- kus  $p$  ja  $q_1, \dots, q_n$  tähistavad *literaale*

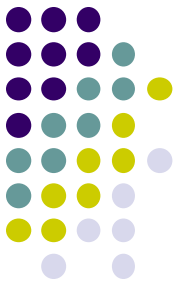
- *disjunktiivne* kuju (sisaldab  $\leq 1$  positiivse literaali):

$$p \vee \neg q_1 \vee \dots \vee \neg q_n$$

- Implikatiivne ja disjunktiivne kuju on *semantiliselt ekvivalentsed*

- *Literaali* – atomaarne valem või selle eitus

- *Atomaarne valem*:  $P(t_1, \dots, t_n)$ , kus  $P$  – predikaadi sümbol  
 $t_1, \dots, t_n$  – termid



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

- Operatsiooniline semantika:

*"Programmi  $p$  täitmiseks tuleb täita alamprogrammid  $q_1, \dots, q_n$ "*

- Deklaratiivne semantika:

*"Lause  $p$  tõesus järeldeb lausete  $q_1, \dots, q_n$  tõesusest"* ehk

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

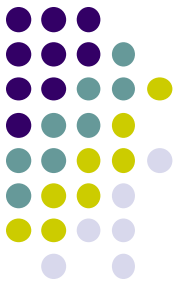
- Muutujate (vaikimisi) tõlgendamine reeglis:

$p(X_1, \dots, X_n) :- q_1(Y_1, \dots, Y_m), \dots, q_n(Z_1, \dots, Z_l)$  (Prolog)

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

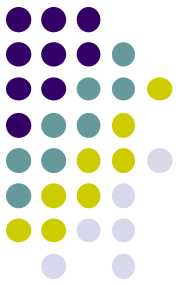
kõik muutujad Horni lauses on kvantoritega seotud





# 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:  $?- \text{isa}(\text{juku}, X).$



# Suletud maailma eeldus

## Suletud maailma eeldus:

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

- **NB!**

- Faktide puudumine teadmusbaasist ei tähenda määramatust
- Fakti puudumine teadmusbaasist tähendab **fakti eitust!**



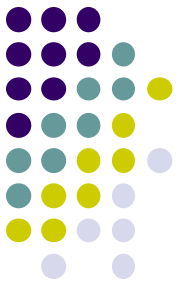


# Resolutsioon

- *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  $\Delta$  on reegli keha.

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

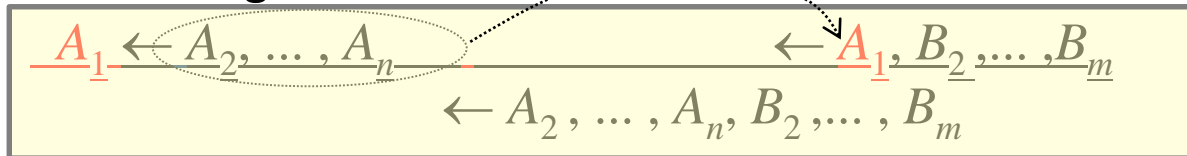


# Resolutsiooni reegel

Olgu Horni lause (HL) implikatsiooni kujul

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

siis resolutsiooni reeglit

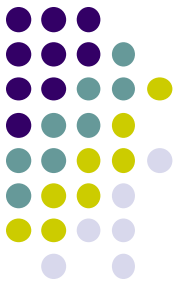


rakendades asendame päringus literaali  $A_1$  reegli  $A_1 \leftarrow A_2, \dots, A_n$  põhjal eeldustega  $A_2, \dots, A_n$ .

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

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

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

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

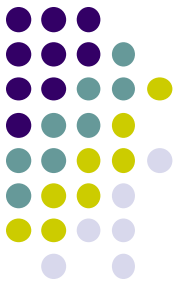
Olgu  $\Gamma \equiv \neg A_1 \vee A_2 \vee \dots \vee A_n$  ja  $\Delta \equiv A_1 \vee B_2 \vee \dots \vee B_m$

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

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

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

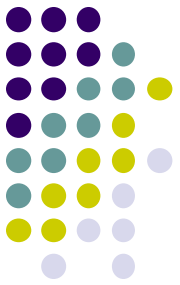
saame  $A_2 \vee \dots \vee A_n \vee B_2 \vee \dots \vee B_m$



# Faktoriseerimisreegel (abireegel)

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

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



# 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 $\Gamma$ :

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).

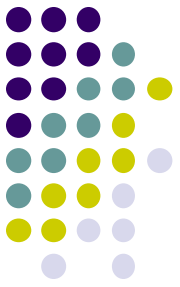
$\neg$  isa(X, Y)  $\vee$   $\neg$  ema(Y, Z)  $\vee$  vanaisa(X, Z).

ja valem **A**, mida tahame tõestada hulgal  $\Gamma$ , kus

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

Seega näitame, et kehtib  $\Gamma \rightarrow \mathbf{A}$ :

# 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  $\Gamma$  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  $\Gamma'$ -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  $\neg \text{isa}(X, Y) \vee \neg \text{isa}(Y, Z) \vee \text{vanaisa}(X, Z)$ .

konkretiseeringud on:

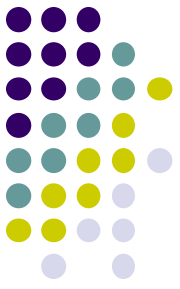
1)  $\neg \text{isa}(\text{leo}, \text{jaan}) \vee \neg \text{isa}(\text{jaan}, \text{martin}) \vee \text{vanaisa}(\text{leo}, \text{martin})$ .

2)  $\neg \text{isa}(\text{leo}, \text{peeter}) \vee \neg \text{isa}(\text{peeter}, \text{martin}) \vee \text{vanaisa}(\text{leo}, \text{martin})$ .

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



# Resolutsioon muutujaid sisaldavate Horni lausetega



- Kuidas leida konkretiseering, mida kasutades saab rakendada resolutsiooni?
- Olgu reeglis

$$\frac{\neg \underline{A_1} \vee A_2 \vee \dots \vee A_n \quad \underline{A'_1} \vee B_2 \vee \dots \vee B_m}{A_2 \vee \dots \vee A_n \vee B_2 \vee \dots \vee B_m}$$

- literaalid  $A_1$  ja  $A'_1$ , mis erinevad ainult nendes esinevate muutujate nimede poolest.
- Tuletame meelde muutujate interpretatsiooni reeglis:  
 $\forall x_1 \dots x_n \exists y_1 \dots y_m z_1 \dots z_l : P(x_1 \dots x_n) \leftarrow Q_1(y_1 \dots y_m) \wedge \dots \wedge Q_n(z_1 \dots z_l)$
- 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



Näide (järg - unifitseerimine):

Et rakendada resolutsioonireeglit lausetele 2 ja 7, kus

2. `isa(jaan, martin)`

7.  $\neg \text{isa}(X, Y) \vee \neg \text{isa}(Y, Z) \vee \text{vanaisa}(X, Z)$ ,

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

Muutujate asendus e. substitutsioon:  $\{x_1/t_1, \dots, x_n/t_n\}$ , kus

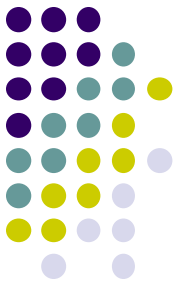
$x_i$  – asendatav muutuja

$t_i$  – asendav term, kus  $t_i$  on erinev  $x_i$  –st

Selgitus: Näites kasutame asendust  $\{X/\text{jaan}, Y/\text{martin}\}$

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

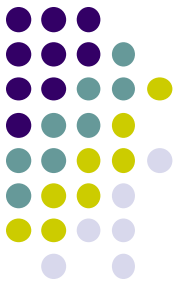
$7 \xrightarrow{\text{unifitseerimine}} 7'$  :  $\neg \text{isa}(\text{jaan}, \text{martin}) \vee \neg \text{isa}(\text{martin}, Z) \vee \text{vanaisa}(\text{jaan}, Z)$ .



# Termide unifitseerimine

Definitsioon (Termide unifitseeruvus):

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



# Termide unifitseerimine

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

Peale  $X$  ja  $Y$  unifitseerimist ja RR reegli rakendamist, saame

7".  $\neg \text{isa}(\text{martin}, Z) \vee \text{vanaisa}(\text{jaan}, Z)$ .

Unifitseerime 3 ja 7":

3.  $\text{isa}(\text{martin}, \text{veiko})$ .

7".  $\neg \text{isa}(\text{martin}, Z) \vee \text{vanaisa}(\text{jaan}, Z)$ .

Unifitseerija  $\{Z/\text{veiko}\}$  annab

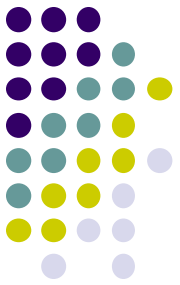
7"".  $\neg \text{isa}(\text{martin}, \text{veiko}) \vee \text{vanaisa}(\text{jaan}, \text{veiko})$

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

9.  $\neg \text{vanaisa}(\text{jaan}, \text{veiko})$

10.  $\text{vanaisa}(\text{jaan}, \text{veiko})$

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



# Termide unifitseerimine

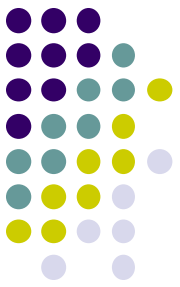
- Kuidas unifitseerida, kui unifitseeritavad termid sisaldavad muutujaid?

*Minimaalse substituutsiooni reegel:*

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

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



# Kõige üldisem unifitseerija - mgu

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

Definitsioon (Kõige üldisem unifitseerija - mgu):

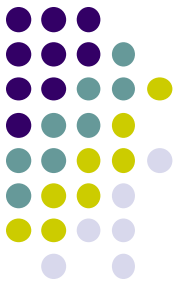
Termide  $t_1$  ja  $t_2$  kõige üldisem unifitseerija on asendus  $\rho$ , mis rahuldab tingimusi:

1.  $\rho$  on termide  $t_1$  ja  $t_2$  unifitseerija
2.  $t_1$  ja  $t_2$  iga teise unifitseerija  $\sigma$  korral leidub veel asendus  $\tau$ , 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

- $\text{mgu}(P(a,X), P(Y,b)) = \{Y/a, X/b\}$
- $\text{mgu}(P(X, f(X)), P(f(Y), U)) = \{X/f(Y), U/f(f(Y))\}$
- $\text{mgu}(L(g(X,X)), L(g(f(a), f(a)))) = \{X/f(a)\}$
- $\text{mgu}(R(a,b), R(a,b)) = \{\}$

# Resolutsioon unifitseeritavatel literalidel



Unifitseerimisega resolutsiooni reegel:

$$\frac{\neg A_1 \vee A_2 \vee \dots \vee A_n \quad B_1 \vee B_2 \vee \dots \vee B_m \quad \sigma = \text{mgu}(A_1, B_1)}{(A_2 \vee \dots \vee A_n \vee B_2 \vee \dots \vee B_m)\sigma}$$

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



# Kõige üldisem unifitseerija - mgu



## Lause:

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

- Lause tõestuseks esitame järgnevalt mgu-algoritmi

# Algoritm mgu leidmiseks (1)

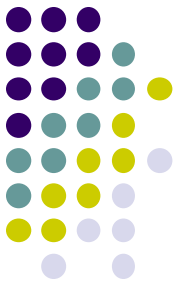


```
substitution mgu(x,y)
term x, y;                                     % Olgu x ja y unifitseeritavad termid
{ int i;
  Substitutions list g;                         % Asendused salvestame listi g

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

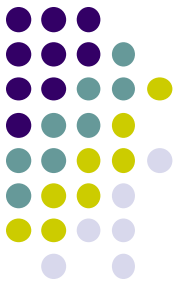
if x==y return EMPTYSUBST;                     % kui X ja Y on ühesugused
else if variable(x) return mgu_var(x,y);      % Kui x on muutuja
else if variable(y) return mgu_var(y,x);      % Kui y on muutuja
else if (constant(x) || constant(y)) return NOSUBST;
                                                % 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;
}
```

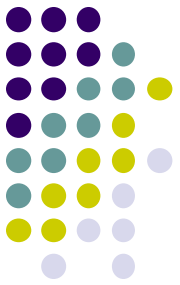


### Case 3: Muutujate asendatavuse kontroll

```
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:

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



# Unifitseerimise harjutusi

Missugused alltoodud asendustest on lubatud?

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

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:

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

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