Prologin resoluutiosta

Horn-klausuuleja ovat

Ohjelma-	$p(X)$	$\forall x\ P(x)$
klausuulit	$r(X,Y)$:- $q(X)$	$\forall x \forall y (R(x,y) \vee \neg Q(x))$
Kyselyt	:- $q(X), s(X)$	$\forall x (\neg Q(x) \vee \neg S(x))$

Lineaarinen input-resoluutio

Olkoon P klausuulijoukko ja G kysely. Vieressä oleva on refutaatio *lineaarisella input-resoluutiolla* (LI), jos $G_0 = G$ ja kukin C_i on P:n klausuuli (muuttujat uudelleennimetty).

$$
\begin{tikzcd}
G_0 \arrow[d] & C_0 \arrow[dl] \\
G_1 \arrow[d] & C_1 \arrow[dl] \\
G_2 \arrow[d] & C_2 \arrow[dl] \\
G_3 & \\
\vdots & \\
G_n \arrow[d] & C_n \arrow[dl] \\
\square &
\end{tikzcd}
$$

Väite. Olkoon P joukko ohjelmaklausuuleja, G kysely ja $P \cup \{G\}$ toteutumaton. Tällöin on olemassa $P \cup \{G\}$:n refutaatio lineaarisella input-resoluutiolla siten, että refutaatio alkaa klausuulista $C_0 = G$.

Todistus: Lineaarinen resoluutio takaa refutaation olemassaolon (T II.14.4). Klausuuleissa G_i on vain negatiivisia literaaleja, joten jokaisella askeleella resolventin C_i tulee olla ohjelmaklausuuli (jossa on siis täsmälleen yksi positiivinen literaali).

Järjestetyt Horn-klausuulit

$$
\begin{tikzcd}
G_0 \arrow[d] & C_0 \arrow[dl] \\
G_1 \arrow[d] & C_1 \arrow[dl] \\
G_2 \arrow[d] & C_2 \arrow[dl] \\
G_3 & \\
\vdots & \\
G_n \arrow[d] & C_n \arrow[dl] \\
\square &
\end{tikzcd}
$$

Olkoot

$G = \{\neg A_0, \ldots, \neg A_n\}$ ja

$C = \{B, \neg B_0, \ldots, \neg B_m\}$

järjestettyjä klausuuleita, ja θ literaalien A_i ja B yleisin unifioija. Klausuuleista G ja C saadaan resoluutiolla järjestetty klausuuli $G = \{\neg A_0, \ldots, \neg A_{i-1}, \neg B_0, \ldots, \neg B_m, \neg A_{i+1}, \ldots, \neg A_n\}\theta$.

Olkoon P joukko järjestettyjä klausuuleita ja G järjestetty kysely. Vieressä oleva on refutaatio *LD-resoluutiolla* (Linear Definite), jos $G_0 = G$ ja kukin C_i on P:n klausuuli (muuttujat uudelleennimetty).

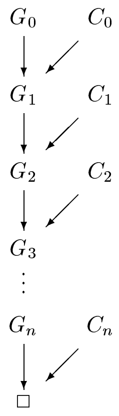
LD-resoluutio valintasäännöllä = SLD-resoluutio

Valintasääntö on funktio R, joka valitsee järjestetystä ei-tyhjästä klausuulista C literaalin $L = R(C)$.

Prologin valintasääntö R_P:

Klausuulille $C = \{\neg A_1, \ldots, \neg A_n\}$, $R_P(C) = \neg A_1$.

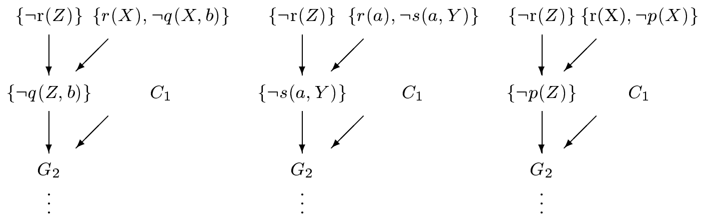
Olkoon P joukko järjestettyjä klausuuleita ja G järjestetty kysely. Vieressä oleva on refutaatio *SLD-resoluutiolla* (Linear Definite), jos $G_0 = G$ ja kukin C_i on P:n klausuuli (muuttujat uudelleennimetty), ja *askeleella i resoluutiosääntöä sovellettiin klausuuleihin* G_i *ja* C_i *literaaliin* $R(G_i)$.



Prologin hakumekanismi

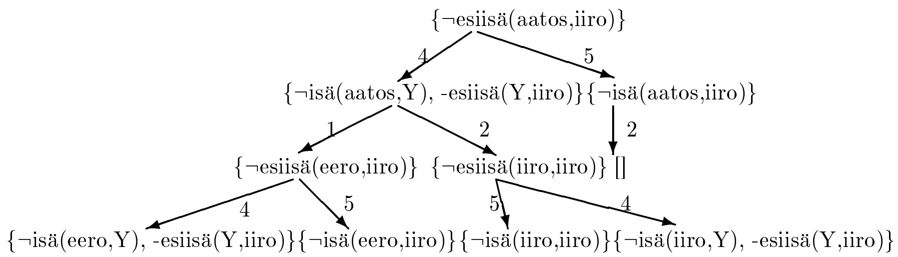
```
r(X) :- q(X,b).
r(a) :- s(a,Y).
r(X) :- p(X).
...
```

Kysely: r(Z)




```
1:  isä(aatos,eero).
2:  isä(aatos,iiro).
3:  isä(oiva,aatos).
4:  esiisä(X,Z) :- isä(X,Y), esiisä(Y,Z).
5:  esiisä(X,Y) :- isä(X,Y).
```

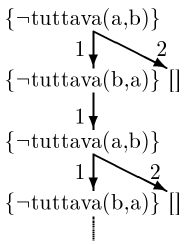
Kyselyn `esiisä(aatos,iiro)` hakupuu:



Depth-first haku: epätäydellisyys

```
1:  tuttava(X,Y) :- tuttava(Y,X)
2:  tuttava(a,b).
```

Kysely: tuttava(a,b) ?



Prologin not-operaattori

- Halutaan käyttää muunkinlaisia sääntöjä kuin $P(t_1, \ldots, t_n)$ ja $P(t_1, \ldots, t_n)$:- $P_1(s_{11}, \ldots, s_{1n_1}), \ldots, P_m(s_{m1}, \ldots, s_{mn_m})$. Esim. ei-operaatiota säännön ehto-osassa.
- Koko predikaattilogiikka liian monimutkainen: ohjelmien suorittaminen ei olisi kyllin tehokasta
- Laajennetaan resoluutioproseduuria:
  - Tulokset käytännön sovelluksissa useimmiten haluttuja
  - Tulokset logiikan kannalta usein/joskus virheellisiä

NOT – negaatio epäonnistumisena

"Jos ei voida päätellä ϕ, päätellään not ϕ"

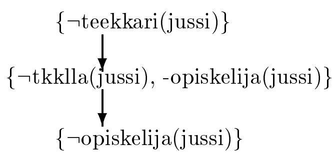
```
koululainen(oskari).
koululainen(emilia).
sairas(oskari).
koulussa(X) :- koululainen(X), not sairas(X).
```

Kysely: koulussa(oskari) ?
Kysely: koulussa(emilia) ?

Määritelmä. *Äärellinen epäonnistunut SLD-puu* P:lle G:lle ja R:lle on äärellinen SLD-resoluution hakupuu, jonka mikään lehti ei ole tyhjä klausuuli $\Box$.

Äärellinen epäonnistunut SLD-puu kyselylle `teekkari(jussi)`:

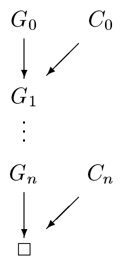
```
teekkari(X) :- tkklla(X),
               opiskelija(X).
opiskelija(teemu).
tkklla(jussi).
tkklla(teemu).
```



SLDNF-resoluutio

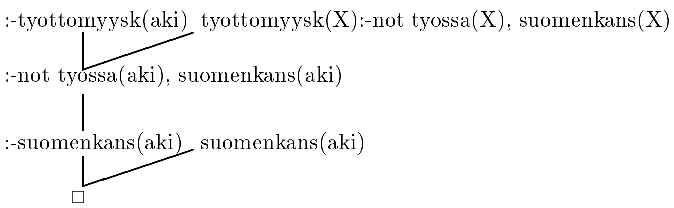
Olkoon P joukko järjestettyjä klausuuleita ja G järjestetty kysely. Vieressä oleva on refutaatio *SLDNF-resoluutiolla* (Selection function, Linear Definite, Negation as Failure), jos $G_0 = G$ ja kukin C_i on P:n klausuuli (mahdollisesti muuttujat uudelleennimettynä), ja

- $R(G_i)$ on not L, L on muuttujaton, ja L:llä P:llä ja R:llä on äärellinen epäonnistunut SLDNF-puu ja G_{i+1} on $G_i - \{\text{not } L\}$, tai
- askeleella i suoritettiin resoluutio klausuuleille G_i ja C_i literaalilla $R(G_i)$.




```
suomenkans(aki).
suomenkans(eve).
tyossa(eve).
tyottomyysk(X) :- not(tyossa(X)), suomenkans(X).
```

Kyselylle `tyottomyysk(aki)` saadaan seuraava SLDNF-refutaatio:



Prolog-ohjelman täydentäminen

$$P(t_{11}, \ldots, t_{1n}) \text{ :- } l_{11}, \ldots, l_{1m_1}$$
$$\vdots$$
$$P(t_{r1}, \ldots, t_{rn}) \text{ :- } l_{r1}, \ldots, l_{rm_r}$$

P-predikaatin klausuulit täydenetään lauseeksi

$$\begin{aligned}
&\forall x_1 \cdots \forall x_n \ (P(x_1, \ldots, x_n) \leftrightarrow \\
&\quad (\exists y_{11} \cdots \exists y_{1v_1} (x_1 = t_{11} \wedge \cdots \wedge x_n = t_{1n} \wedge l_{11} \wedge \cdots \wedge l_{1m_1}) \vee \\
&\quad \cdots \\
&\quad \vee \exists y_{r1} \cdots \exists y_{rv_r} (x_1 = t_{r1} \wedge \cdots \wedge x_n = t_{rn} \wedge l_{r1} \wedge \cdots \wedge l_{rm_r})))
\end{aligned}$$

Ohjelman P vastaava lausejoukko Comp(P) sisältää täydennettyjen klausuulien lisäksi yksikäsitteisten nimien aksioomat.

Ohjelma

```
suomenkansalainen(aki).
suomenkansalainen(eve).
tyossa(eve).
tyottomyyskorvausta(X) :- not(tyossa(X)),
                          suomenkansalainen(X).
```

on täydennettynä

$$\forall x \ (\text{suomenkansalainen}(x) \leftrightarrow (x = \text{aki} \vee x = \text{eve}))$$
$$\forall x \ (\text{tyossa}(x) \leftrightarrow (x = \text{eve}))$$
$$\forall x \ (\text{tyottomyyskorvausta}(x) \leftrightarrow$$
$$(\neg \text{ tyossa}(x) \wedge \text{ suomenkansalainen}(x)))$$
$$\neg(\text{aki} = \text{eve})$$

Väite. Jos not-operaatiota sisältävällä ohjelmalla P ja kyselyllä G $= \{\neg L_1, \ldots, \neg L_m\}$ on SLDNF-refutaatio, niin Comp(P) $\models \exists x_1 \cdots \exists x_n (L_1 \wedge \cdots \wedge L_m)$.

Esimerkki. Kaikkia Comp(P):n loogisia seurauksia ei saada SLDNF-resoluutiolla:

`v(X) :- not(q(X)).`	$\forall x (v(x) \leftrightarrow \neg q(x))$
`q(a) :- not r(a).`	$q(a) \leftrightarrow \neg r(a)$
`r(a) :- p(a).`	$r(a) \leftrightarrow (p(a) \vee \neg p(a))$ eli $r(a)$
`r(a) :- not p(a).`	
`p(X) :- p(f(X)).`	$\forall x (p(x) \leftrightarrow p(f(x)))$

Kysely: `v(a)`?

Jos negaation yhteydessä käytetään muuttujia sisältäviä literaaleja, SDLNF-resoluution korrektius menetetään:

```
koululainen(oskari).
sairas(liisa).
koulussa(X) :- not sairas(X), koululainen(X).
```

Kysely: `koulussa(X)` ?

$\forall x (\text{koululainen}(x) \leftrightarrow x = \text{oskari})$

$\forall x (\text{sairas}(x) \leftrightarrow x = \text{liisa})$

$\forall x (\text{koulussa}(x) \leftrightarrow (\neg \text{sairas}(x) \wedge \text{koululainen}(x)))$

$\neg (\text{oskari} = \text{liisa})$