Tietämyksen esittäminen predikaattilogiikalla

- Tietämystä voidaan esittää valitsemalla sopiva predikaattilogiikan lausejoukko $\Sigma \subseteq \mathcal{L}$, joka muodostaa *eksplisiittisen* tietämyksen. Lausejoukon Σ loogiset seuraavuudet muodostavat *implisiittisen* tietämyksen: väittämät, jotka voidaan päätellä annetusta tietämyksestä.

Esimerkki.

$$\begin{aligned}\Sigma \;=\; \{ & \forall x \forall y \forall z (\mathrm{johto}(x,y) \land \mathrm{jännite}(x,z) \rightarrow \mathrm{jännite}(y,z)), \\ & \mathrm{johto}(\mathrm{p1},\mathrm{p2}),\ \mathrm{johto}(\mathrm{p2},\mathrm{p3}),\ \mathrm{jännite}(\mathrm{p1},220)\ \}\end{aligned}$$

Nyt esim. lause $\mathrm{jännite}(\mathrm{p3},220)$ on implisiittistä tietämystä.

- Jokainen lausejoukko Σ määrittää joukon *malleja*, eli struktuureja $\mathcal{A}$, joissa lausejoukon kaikki lauseet ovat tosia.
- Lausejoukon Σ mallit puolestaan määräävät lausejoukon loogisten seurauksien joukon $\mathrm{Cn}(\Sigma) = \{\phi \mid \Sigma \models \phi\}$.
- Kaikki struktuurit ovat tyhjän lausejoukon $\emptyset$ malleja. Täten $\mathrm{Cn}(\emptyset)$ on itse asiassa pätevien lauseiden joukko.
- Enemmän lauseita $\Rightarrow$ vähemmän malleja $\Rightarrow$ enemmän loogisia seurauksia: siis $\Sigma_1 \subseteq \Sigma_2 \Rightarrow \mathrm{Cn}(\Sigma_1) \subseteq \mathrm{Cn}(\Sigma_2)$.
- *Tietämyksen esittäminen*: ongelmana rajoittaa mallien joukko siten, että saadaan halutut loogiset seuraukset.
- Jos lausejoukko tulee ristiriitaiseksi, sillä ei ole mallia ja kaikki lauseet ovat tällöin lausejoukon loogisia seurauksia.

Lisähuomioita loogisesta seuraavuudesta

- Jos $\Sigma \models \phi$, niin $\mathrm{Cn}(\Sigma) = \mathrm{Cn}(\Sigma \cup \{\phi\})$.

  Lausejoukon loogisia seurauksia ei siis kannata lisätä lausejoukkoon, koska lausejoukon mallit eivät näin muutu!

- Oletetaan, että $\Sigma \not\models \phi$ ja että Σ halutaan laajentaa lausejoukoksi Σ' siten, että $\Sigma' \models \phi$.

  Apuna voidaan käyttää vastamallia/vastamalleja $\mathcal{A}$, joille $\mathcal{A} \models \Sigma$ ja $\mathcal{A} \not\models \phi$: lausejoukkoon Σ lisättävän lauseen ψ tulisi sulkea pois vastamalli/vastamalleja $\mathcal{A}$, ts. $\mathcal{A} \not\models \Sigma \cup \{\psi\}$.

  Lause ϕ toteutaa myös tämän vaatimuksen, mutta lausetta ϕ ei välttämättä haluta liittää eksplisiittiseen tietämykseen.

  Esim. $\Sigma = \{\forall x (P(x) \rightarrow Q(x))\}$, $\phi = Q(a)$ ja $\psi = P(a)$.

Esimerkki. Kieli $\mathcal{L}$: $\mathrm{aviossa}(x,y), \mathrm{nainen}(x), \mathrm{mies}(y)$

U_i	$\mathrm{aviossa}^{\mathcal{A}_i}$	$\mathrm{nainen}^{\mathcal{A}_i}$	$\mathrm{mies}^{\mathcal{A}_i}$
$\{1\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\{1\}$	$\emptyset$	$\emptyset$	$\{1\}$
$\{1\}$	$\emptyset$	$\{1\}$	$\emptyset$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3\}$	$\{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle\}$	$\{1,2\}$	$\{3\}$
$\{1,2,3\}$	$\{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle\}$	$\{1,2\}$	$\{1,3\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3,4,5\}$	$\{\langle 1,2\rangle, \langle 2,1\rangle\}$	$\{1,2,5\}$	$\{1,2,3,4,5\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$$\Sigma_a = \left\{ \begin{array}{l} \neg\exists x(\mathrm{mies}(x) \wedge \mathrm{nainen}(x)) \\ \forall x(\neg(\mathrm{aviossa}(x,x))) \\ \forall x \forall y(\mathrm{aviossa}(x,y) \rightarrow \mathrm{aviossa}(y,x)) \\ \forall x \forall y \forall z(\mathrm{aviossa}(x,y) \wedge \mathrm{aviossa}(x,z) \rightarrow y = z) \\ \forall x \forall y(\mathrm{aviossa}(x,y) \rightarrow \\ \qquad ((\mathrm{nainen}(x) \wedge \mathrm{mies}(y)) \vee (\mathrm{mies}(x) \wedge \mathrm{nainen}(y)))) \end{array} \right\}$$

U_i	$\mathrm{aviossa}^{\mathcal{A}_i}$	$\mathrm{nainen}^{\mathcal{A}_i}$	$\mathrm{mies}^{\mathcal{A}_i}$
$\{1,2,3\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\{1,2,3\}$	$\{\langle 2,1\rangle,\langle 1,2\rangle\}$	$\{1\}$	$\{2,3\}$
$\{1,2,3\}$	$\{\langle 1,3\rangle,\langle 3,1\rangle\}$	$\{1\}$	$\{2,3\}$
$\{1,2,3\}$	$\emptyset$	$\{1,2,3\}$	$\emptyset$
$\{1,2,3,4,5\}$	$\{\langle 1,2\rangle,\langle 2,1\rangle\}$	$\{1\}$	$\{2\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Yksikäsitteisten nimien aksiomat ja sulkeuma-aksiomat

Kieli $\mathcal{L}$: vakiosymbolit $\mathrm{aki}, \mathrm{ann}, \mathrm{eve}$ ja predikaattisymboleita

Jokaisen vakiosymbolin viitattava eri alkioon (unique names):

$$\neg(\mathrm{ann} = \mathrm{eve}), \neg(\mathrm{aki} = \mathrm{eve}), \neg(\mathrm{aki} = \mathrm{ann})$$

Tarkasteltavassa joukossa ei muita kuin vakiosymbolien nimeämiä alkioita (domain closure):

$$\forall x(x = \mathrm{aki} \vee x = \mathrm{ann} \vee x = \mathrm{eve})$$

U_i	$\mathrm{aki}^{\mathcal{A}_i}$	$\mathrm{ann}^{\mathcal{A}_i}$	$\mathrm{eve}^{\mathcal{A}_i}$
$\{1\}$	1	1	1
$\{1,2\}$	1	1	1
$\{1,2\}$	1	1	2
$\{1,2\}$	1	2	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3\}$	1	1	1
$\{1,2,3\}$	1	1	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3\}$	3	3	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3,\ldots\}$	1	1	1

$$\neg(\mathrm{ann} = \mathrm{eve}), \neg(\mathrm{aki} = \mathrm{eve}), \neg(\mathrm{aki} = \mathrm{ann})$$

U_i	$\mathrm{aki}^{\mathcal{A}_i}$	$\mathrm{ann}^{\mathcal{A}_i}$	$\mathrm{eve}^{\mathcal{A}_i}$
$\{1,2,3\}$	1	2	3
$\{1,2,3\}$	1	3	2
$\{1,2,3\}$	2	1	3
$\{1,2,3\}$	2	3	1
$\{1,2,3\}$	3	1	2
$\{1,2,3\}$	3	2	1
$\{1,2,3,4\}$	1	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$$\forall x(x = \text{aki} \lor x = \text{ann} \lor x = \text{eve})$$

U_i	$\text{aki}^{\mathcal{A}_i}$	$\text{ann}^{\mathcal{A}_i}$	$\text{eve}^{\mathcal{A}_i}$
$\{1\}$	1	1	1
$\{1,2\}$	1	1	2
$\{1,2\}$	1	2	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3\}$	1	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3\}$	3	2	1

$$\Sigma_u = \left\{ \begin{array}{l} \neg(\text{ann} = \text{eve}), \neg(\text{aki} = \text{eve}), \neg(\text{aki} = \text{ann}) \\ \forall x(x = \text{aki} \lor x = \text{ann} \lor x = \text{eve}) \end{array} \right\}$$

U_i	$\text{aki}^{\mathcal{A}_i}$	$\text{ann}^{\mathcal{A}_i}$	$\text{eve}^{\mathcal{A}_i}$
$\{1,2,3\}$	1	2	3
$\{1,2,3\}$	1	3	2
$\{1,2,3\}$	2	1	3
$\{1,2,3\}$	2	3	1
$\{1,2,3\}$	3	1	2
$\{1,2,3\}$	3	2	1

Avioliittoesimerkki jatkuu...

$$\Sigma = \Sigma_a \cup \Sigma_u \cup \{\forall x(\text{nainen}(x) \lor \text{mies}(x))\}$$

U_i	$\text{aviossa}^{\mathcal{A}_i}$	$\text{nainen}^{\mathcal{A}_i}$	$\text{mies}^{\mathcal{A}_i}$	$\text{aki}^{\mathcal{A}_i}$	$\text{ann}^{\mathcal{A}_i}$	$\text{eve}^{\mathcal{A}_i}$
$\{1,2,3\}$	$\emptyset$	$\emptyset$	$\{1,2,3\}$	1	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3\}$	$\emptyset$	$\{1,2,3\}$	$\emptyset$	1	2	3
$\{1,2,3\}$	$\emptyset$	$\{2,3\}$	$\{1\}$	1	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{1,2,3\}$	$\{\langle 1,2\rangle, \langle 2,1\rangle\}$	$\{2,3\}$	$\{1\}$	1	2	3
$\{1,2,3\}$	$\{\langle 1,3\rangle, \langle 3,1\rangle\}$	$\{2,3\}$	$\{1\}$	1	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$$\begin{aligned} \Sigma = {} & \Sigma_a \cup \Sigma_u \cup \{\forall x(\text{nainen}(x) \lor \text{mies}(x)), \text{mies}(\text{aki}), \\ & \text{nainen}(\text{ann}), \text{nainen}(\text{eve})\} \end{aligned}$$

Lausejoukolla Σ on *olennaisesti* seuraavat mallit (struktuurit, joissa kaikki lausejoukon Σ lauseet ovat tosia):

U_i	$\text{aviossa}^{\mathcal{A}_i}$	$\text{nainen}^{\mathcal{A}_i}$	$\text{mies}^{\mathcal{A}_i}$	$\text{aki}^{\mathcal{A}_i}$	$\text{ann}^{\mathcal{A}_i}$	$\text{eve}^{\mathcal{A}_i}$
$\{1,2,3\}$	$\emptyset$	$\{2,3\}$	$\{1\}$	1	2	3
$\{1,2,3\}$	$\{\langle 1,2\rangle, \langle 2,1\rangle\}$	$\{2,3\}$	$\{1\}$	1	2	3
$\{1,2,3\}$	$\{\langle 1,3\rangle, \langle 3,1\rangle\}$	$\{2,3\}$	$\{1\}$	1	2	3

1. Onko $\Sigma \cup \{\text{aviossa}(\text{aki}, \text{ann})\} \models \text{aviossa}(\text{ann}, \text{aki})$?
2. Onko $\Sigma \cup \{\text{aviossa}(\text{aki}, \text{ann})\} \models \neg\text{aviossa}(\text{eve}, \text{aki})$?
3. Onko $\Sigma \models \neg\text{aviossa}(\text{eve}, \text{ann})$?
4. Onko $\Sigma \models \neg\text{aviossa}(\text{aki}, \text{ann})$?

Lisää tietämyksen esittämisestä...

Kieli $\mathcal{L}$: vakiot aki, ann, eve ja predikaattisymboli pitää

$$\Sigma = \left\{ \begin{array}{l} \forall x(x = \text{aki} \vee x = \text{ann} \vee x = \text{eve}) \\ \neg(\text{aki} = \text{ann}), \neg(\text{aki} = \text{eve}), \neg(\text{ann} = \text{eve}) \end{array} \right\}$$

i	U_i	$\text{pitää}^{\mathcal{A}_i}$	$\text{aki}^{\mathcal{A}_i}$	$\text{ann}^{\mathcal{A}_i}$	$\text{eve}^{\mathcal{A}_i}$
0	$\{1,2,3\}$	$\emptyset$	1	2	3
1	$\{1,2,3\}$	$\{\langle 1,1\rangle\}$	1	2	3
2	$\{1,2,3\}$	$\{\langle 1,2\rangle\}$	1	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
511	$\{1,2,3\}$	$\{\langle 1,1\rangle, \langle 1,2\rangle, \ldots\}$ $= U_{511} \times U_{511}$	1	2	3

$$\Sigma' = \Sigma \cup \{\text{pitää}(\text{aki}, \text{ann}), \text{pitää}(\text{eve}, \text{eve})\}$$

i	U_i	$\text{pitää}^{\mathcal{A}_i}$	$\text{aki}^{\mathcal{A}_i}$	$\text{ann}^{\mathcal{A}_i}$	$\text{eve}^{\mathcal{A}_i}$
2	$\{1,2,3\}$	$\{\langle 1,2\rangle, \langle 3,3\rangle\}$	1	2	3
3	$\{1,2,3\}$	$\{\langle 1,1\rangle, \langle 1,2\rangle, \langle 3,3\rangle\}$	1	2	3
4	$\{1,2,3\}$	$\{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 3,3\rangle\}$	1	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
511	$\{1,2,3\}$	$U_{511} \times U_{511}$	1	2	3

Onko $\Sigma' \models \neg\text{pitää}(\text{aki}, \text{eve})$?

Haluttu lausejoukko on itse asiassa

$$\begin{aligned} \Gamma_2 \;=\; & \{\text{pitää}(\text{aki}, \text{ann}), \text{pitää}(\text{eve}, \text{eve}), \\ & \neg\text{pitää}(\text{aki}, \text{aki}), \neg\text{pitää}(\text{aki}, \text{eve}), \neg\text{pitää}(\text{ann}, \text{aki}), \\ & \neg\text{pitää}(\text{ann}, \text{ann}), \neg\text{pitää}(\text{ann}, \text{eve}), \neg\text{pitää}(\text{eve}, \text{aki}), \\ & \neg\text{pitää}(\text{eve}, \text{ann})\} \end{aligned}$$

"Negatiivista tietoa" saattaa olla PALJON enemmän kuin "positiivista" ...

Lausejoukko

$$\Gamma = \{\text{pitää}(\text{aki}, \text{ann}), \text{pitää}(\text{eve}, \text{eve})\}$$

voidaan antaa identiteettipredikaattia käyttäen myös joukkona

$$\begin{aligned}\{\forall x \forall y(((x = \text{aki}) \land (y = \text{ann})) \lor \\ ((x = \text{eve}) \land (y = \text{eve})) \rightarrow \text{pitää}(x, y))\}\end{aligned}$$

Rajoitetaan pitäminen siten, ettei pitämistä tapahdu ellei Γ:ssa asiasta ole erikseen mainittu:

$$\begin{aligned}\Gamma' = \{\forall x \forall y(((x = \text{aki}) \land (y = \text{ann})) \lor \\ ((x = \text{eve}) \land (y = \text{eve})) \leftrightarrow \text{pitää}(x, y))\}\end{aligned}$$

Lausejoukon

$$\Sigma'' = \Sigma \cup \Gamma'$$

kaikki struktuurit ovat sellaisia, että vain aki pitää annista ja eve pitää itsestään:

i	U_i	$\text{pitää}^{\mathcal{A}_i}$	$\text{aki}^{\mathcal{A}_i}$	$\text{ann}^{\mathcal{A}_i}$	$\text{eve}^{\mathcal{A}_i}$
2	$\{1, 2, 3\}$	$\{\langle 1, 2\rangle, \langle 3, 3\rangle\}$	1	2	3

Onko $\Sigma'' \models \neg\text{pitää}(\text{aki}, \text{eve})$?

Automaattinen teoreemantodistin OTTER

Syntaksi:	$(\forall x\ \varphi)$	`(all x (`φ`))`
	$(\exists x\ \varphi)$	`(exists x (`φ`))`
	$(\varphi \land \psi)$	`(`φ `&` ψ`)`
	$(\varphi \lor \psi)$	`(`φ `|` ψ`)`
	$(\varphi \rightarrow \psi)$	`(`φ `->` ψ`)`
	$(\varphi \leftrightarrow \psi)$	`(`φ `<->` ψ`)`
	$(t_1 = t_2)$	`(`t_1 `=` t_2`)`
	$(\neg\varphi)$	`(-`φ`)`

Esim.

$\forall x \exists y\ P(x, y)$	`(all x (exists y P(x,y))).`
$\forall x(P(x) \land (Q(x) \lor R(x)))$	`(all x (P(x) & (Q(x) | R(x)))).`

Suositeltavat asetukset

```
set(auto).

formula_list(usable).

% tähän tulee sitten tarvittava lausejoukko

% ja kyselyn negaatio (vain yksi kysely kerrallaan)

end_of_list.
```

Esimerkki. (Avioliittoesimerkki OTTERilla)

```
set(auto).
formula_list(usable).

(all x (nainen(x) | mies(x))).
-(exists x (mies(x) & nainen(x))).

(all x -(aviossa(x,x))).
(all x y (aviossa(x,y) -> aviossa(y,x))).
(all x y z (aviossa(x,y) & aviossa(x,z) -> y=z)).
(all x y (aviossa(x,y) ->
          (nainen(x) & mies(y)) | (mies(x) & nainen(y)) )).

-(ann=eve). -(aki=eve). -(aki=ann).
(all x (x=aki | x=ann | x=eve)).
nainen(ann). nainen(eve). mies(aki). aviossa(aki,ann).

-(-aviossa(eve,aki)).     % !!! Kyselyn _negaatio_ !!!
end_of_list.
```


OTTERin tulosteet

```
usable clausifies to:   list(usable).
                        0 [] nainen(x)|mies(x).
                        0 [] -mies(x)| -nainen(x).
                        0 [] -aviossa(x,x).
                        0 [] -aviossa(x,y)|aviossa(y,x).
                        0 [] -aviossa(x,y)| -aviossa(x,z)|y=z.
                        0 [] -aviossa(x,y)|nainen(x)|mies(x).
                        0 [] -aviossa(x,y)|nainen(x)|nainen(y).
                        0 [] -aviossa(x,y)|mies(y)|mies(x).
                        0 [] -aviossa(x,y)|mies(y)|nainen(y).
                        0 [] ann!=eve.
                        0 [] aki!=eve.
                        0 [] aki!=ann.
                        0 [] x=aki|x=ann|x=eve.
                        0 [] nainen(ann).
                        0 [] nainen(eve).
                        0 [] mies(aki).
                        0 [] aviossa(aki,ann).
                        0 [] aviossa(eve,aki).
                        end_of_list.
```



```
------> process usable:
** KEPT (pick-wt=4): 1 [] -mies(x)| -nainen(x).
** KEPT (pick-wt=3): 2 [] -aviossa(x,x).
** KEPT (pick-wt=6): 3 [] -aviossa(x,y)|aviossa(y,x).
** KEPT (pick-wt=9): 4 [] -aviossa(x,y)| -aviossa(x,z)|y=z.
** KEPT (pick-wt=7): 5 [] -aviossa(x,y)|nainen(x)|mies(x).
** KEPT (pick-wt=7): 6 [] -aviossa(x,y)|nainen(x)|nainen(y).
** KEPT (pick-wt=7): 7 [] -aviossa(x,y)|mies(y)|mies(x).
** KEPT (pick-wt=7): 8 [] -aviossa(x,y)|mies(y)|nainen(y).
** KEPT (pick-wt=3): 10 [copy,9,flip.1] eve!=ann.
** KEPT (pick-wt=3): 12 [copy,11,flip.1] eve!=aki.
** KEPT (pick-wt=3): 14 [copy,13,flip.1] ann!=aki.

------> process sos:
** KEPT (pick-wt=4): 16 [] nainen(x)|mies(x).
** KEPT (pick-wt=9): 17 [] x=aki|x=ann|x=eve.
** KEPT (pick-wt=2): 18 [] nainen(ann).
** KEPT (pick-wt=2): 19 [] nainen(eve).
** KEPT (pick-wt=2): 20 [] mies(aki).
** KEPT (pick-wt=3): 21 [] aviossa(aki,ann).
** KEPT (pick-wt=3): 22 [] aviossa(eve,aki).
16 back subsumes 8.
16 back subsumes 5.
```



```
=========== start of search ===========

given clause #1: (wt=4) 16 [] nainen(x)|mies(x).
given clause #2: (wt=2) 18 [] nainen(ann).
given clause #3: (wt=2) 19 [] nainen(eve).
given clause #4: (wt=2) 20 [] mies(aki).
given clause #5: (wt=3) 21 [] aviossa(aki,ann).
given clause #6: (wt=9) 17 [] x=aki|x=ann|x=eve.
given clause #7: (wt=3) 22 [] aviossa(eve,aki).
given clause #8: (wt=3) 26 [hyper,21,15] ann=ann.
given clause #9: (wt=3) 27 [hyper,21,3] aviossa(ann,aki).
given clause #10: (wt=3) 30 [hyper,17,12,unit_del,10] eve=eve.
given clause #11: (wt=8) 23 [para_into,18.1.1,4.3.1]
                              nainen(x)|-aviossa(y,ann)|-aviossa(y,x).
given clause #12: (wt=3) 31 [para_from,17.2.1,21.1.1,unit_del,2,12]
                              aki=aki.
given clause #13: (wt=3) 32 [hyper,22,3] aviossa(aki,eve).

--> UNIT CONFLICT at   0.06 sec --> 40 [binary,38.1,10.1] $F.
```

```
-------- PROOF --------

3 [] -aviossa(x,y)|aviossa(y,x).
4 [] -aviossa(x,y)| -aviossa(x,z)|y=z.
9 [] ann!=eve.
10 [copy,9,flip.1] eve!=ann.
21 [] aviossa(aki,ann).
22 [] aviossa(eve,aki).
32 [hyper,22,3] aviossa(aki,eve).
38 [hyper,32,4,21,flip.1] eve=ann.
40 [binary,38.1,10.1] $F.

------ end of proof -------

Search stopped by max_proofs option.
```


- Huomaa, että jos OTTER ei löydä todistusta, OTTER voi pysähtyä (ja ilmoittaa "search stopped because sos is empty") tai jäädä ikuiseen silmukkaan. Tämä käyttäytyminen liittyy predikaattilogiikan puoliratkeavuuteen. Esimerkissämme ilmiö tulee esille muuttamalla kyselyksi `-aviossa(aki,ann).`
- Muuttujasidontojen eristämistä varten OTTERissa voi määritellä `$ans`-alkuisia predikatteja, joiden argumenteiksi tulee mielenkiinnon kohteena olevat muuttujat. Esim. kyselyllä `(exists x (-aviossa(aki,x) & $ans(x)))` saadaan todistus

  ```
  2 [] -aviossa(x,x).
  22 [] aviossa(aki,x)| -$ans(x).
  23 [binary,22.1,2.1] -$ans(aki).
  ```

  Tästä tiedämme, että aki on vaatimuksen täyttävä henkilö.

OTTER TKK:n koneissa

- Asennettuna seuraavissa koneissa: alpha, beta, gamma, ...
- Otetaan käyttöön kirjoittamalla: `use otter` (`source /p/setup/use/shell/use.csh`)
- Käynnistäminen: kirjoitetaan lausejoukko tiedostoon xyz.in käynnistetään OTTER kirjoittamalla

  `otter < xyz.in` tai `ott xyz.in`

- Hakemistossa /p/edu/tik-79.144: esimerkkejä (matematiikasta pääasiassa), MS-DOS -binääri, Macintosh-binääri, yms.
- Manuaaliin löytyy linkki kurssin WWW-sivulta