PREDIKAATTILOGIIKKA

1. Predikaattilogiikan kieli
2. Predikaattilogiikan semantiikka
3. Normaalimuodot
4. Semanttiset taulut predikaattilogiikalle
5. Tietämyksen esittämisestä
6. Herbrandin teoreema
7. Unifikaatio
8. Resoluutiosääntö ja -todistukset
9. Ohjelmien oikeellisuustarkastelut

1 Predikaattilogiikan kieli

- Motivaatio
- Predikaattilogiikan aakkosto
- Kielen määritelmä
- Kaavojen muodostaminen
- Kvanttoreihin liittyviä määritelmiä

1.1 Motivaatio

- Lauselogiikka on useisiin tarkoituksiin liian yksinkertainen: olkoon $A =$ "a on viallinen", $B =$ "b on viallinen", $C =$ "c on viallinen".

  Tällöin "kaikki ovat viallisia" $= A \wedge B \wedge C$ ja
  "jokin on viallinen" $= A \vee B \vee C$.

- Erityisesti objektien välisten suhteiden kuvaaminen on hankalaa (tarvitaan paljon lauseita, jotka ovat muodoltaan samankaltaisia).

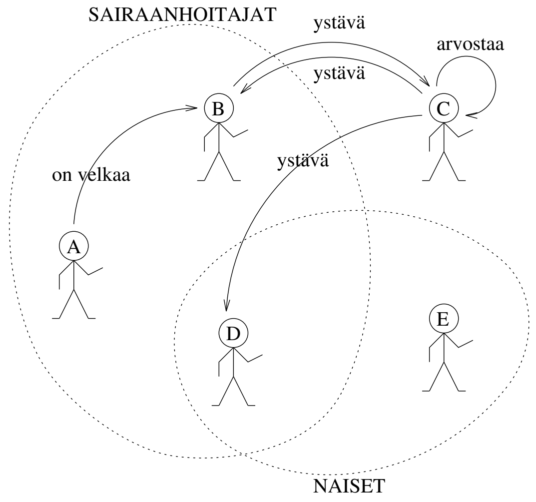
Esimerkki.

"Jos x on isompi kuin y ja y on isompi kuin z,
niin x on isompi kuin z".

$C_d =$ "c on isompi kuin d", $D_e =$ "d on isompi kuin e", ...

$(C_d \wedge D_e \rightarrow C_e) \wedge (C_e \wedge E_d \rightarrow C_d) \wedge (D_e \wedge E_c \rightarrow D_c) \wedge \cdots$

Esimerkki. Alla on kuvattu joitain henkilöiden välisiä suhteita.



1.2 Predikaattilogiikan aakkosto

Predikaattilogiikan kielessä $\mathcal{L}$ käytetään seuraavia symboleja:

- Muuttujasymbolit $\mathcal{V} = \{x, y, z, \ldots\}$
- Vakiosymbolit $\mathcal{C} = \{a, b, c, \ldots\}$
- Funktiosymbolit $\mathcal{F} = \{f, g, h, \ldots\}$
- Predikaattisymbolit $\mathcal{P} = \{=, P, Q, R, \ldots\}$
- Lauselogiikan konnektiivit $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- Kvanttorisymbolit $\exists, \forall$
- Sulut ( ) ja pilkku ,

Määritelmä.

- Jokaisella funktiosymbolilla $f \in \mathcal{F}$ on *paikkaluku* $n > 0$ (joka määrää f:n argumenttien lukumäärän).
- Vastaavasti predikaattisymboleilla $P \in \mathcal{P}$ on paikkaluvut $n \geq 0$.
- Määritellään $\mathcal{F}_n = \{f \in \mathcal{F} \mid f\text{:n paikkaluku on } n\}$, kun $n > 0$, ja $\mathcal{P}_n = \{P \in \mathcal{P} \mid P\text{:n paikkaluku on } n\}$, kun $n \geq 0$.
- Täten $\mathcal{F} = \cup\{\mathcal{F}_n \mid n > 0\}$ ja $\mathcal{P} = \cup\{\mathcal{P}_n \mid n \geq 0\}$.

Huomioita.

- Yhtäsuuruuspredikaatti $= \in \mathcal{P}$ on kaksipaikkainen, eli $= \in \mathcal{P}_2$.
- Vakiosymbolien joukko $\mathcal{C}$ voitaisiin vaihtoehtoisesti määritellä 0-paikkaisten funktiosymbolien joukkona $\mathcal{F}_0$.
- Joukon $\mathcal{P}_0$ symbolit vastaavat lauselogiikan atomisia lauseita.

1.3 Kielen määritelmä

Predikaattilogiikan kielen $\mathcal{L}$ määritelmä on kolmitasoinen: ensin määritellään termit, sitten atomikaavat ja lopulta varsinaiset kaavat.

Määritelmä.

1. Jokainen muuttujasymboli $v \in \mathcal{V}$ on *termi*.
2. Jokainen vakiosymboli $c \in \mathcal{C}$ on *termi*.
3. Jos $f \in \mathcal{F}_n$ on n-paikkainen funktiosymboli ja $t_1, \ldots, t_n$ ovat termejä, niin myös $f(t_1, \ldots, t_n)$ on *termi*.
4. Muita termejä ei ole.

Esimerkki. x, c, $f(x)$, $f(f(f(f(f(x)))))$, $g(f(x), g(f(x), g(x, c)))$.

Määritelmä. Termi, jossa ei esiinny muuttujia, on *muuttujaton termi* (engl. *ground term*).

Määritelmä.

1. Jos t_1 ja t_2 ovat termejä, niin $t_1 = t_2$ on *atomikaava*.
2. Jos $P \in \mathcal{P}_0$ on 0-paikkainen predikaattisymboli, niin P on *atomikaava*.
3. Jos $P \in \mathcal{P}_n$ on n-paikkainen predikaattisymboli (missä $n > 0$) ja $t_1, \ldots, t_n$ ovat termejä, niin $P(t_1, \ldots, t_n)$ on *atomikaava*.
4. Muita atomikaavoja ei ole.

Esimerkki. Atomikaavoja ovat mm.

$$P(x_1), \quad Q, \quad x_1 = x_2, \quad g(x_1, x_2) = f(f(c_1)),$$
$$R(c_1, x_1, y_1) \quad \text{ja } S(x_1, c_1, f(x), h(f(x_1), c_1, x_1), x_2, x_2, x_3),$$

mutta esim. $f(R(x), c)$ ei ole atomikaava (eikä edes termi).

Määritelmä.

1. Jokainen atomikaava ϕ on *kaava*.
2. Jos ϕ ja ψ ovat kaavoja ja x on muuttuja, niin myös

   $$(\neg\phi),\ (\phi \vee \psi),\ (\phi \wedge \psi),\ (\phi \rightarrow \psi),\ (\phi \leftrightarrow \psi),\ (\forall x\phi),\ (\exists x\phi)$$

   ovat *kaavoja*.
3. Muita kaavoja ei ole.

Esimerkki. Predikaattilogiikan kaavoja: $P(c)$, $(\forall x(P(x) \rightarrow Q(x)))$, $(\forall x(P(x) \vee (\exists y Q(x,y))))$, $(\exists x(\forall y(\forall z P(x,y,z))))$.

Symbolijoukkoihin $\mathcal{V}, \mathcal{C}, \mathcal{F}$ ja $\mathcal{P}$ perustuva predikaattilogiikan *kieli* $\mathcal{L}$ määritellään edellä annetuilla periaatteilla muodostettavissa olevien kaavojen joukkona.

Sopimukset sulkeiden käytöstä

- Konnektiivien presedenssiluokat ovat seuraavat:
  1. $\neg$, $\forall v$ ja $\exists v$ (missä $v \in \mathcal{V}$) ovat vahvimmat konnektiivit.
  2. $\vee$ ja $\wedge$ ovat näitä heikompia, mutta vahvempia kuin $\rightarrow$ ja $\leftrightarrow$.
  3. $\rightarrow$ ja $\leftrightarrow$ ovat heikoimmat konnektiivit.
- Lauselogiikan yhteydessä käyttöönotettuja periaatteita sulkeiden vähentämiseksi käytetään myös kaavoja kirjoitettaessa.

Esimerkki. Täten kaava

$$(\exists x(\forall y((\exists z(P(x,z) \wedge P(z,y))) \rightarrow ((Q(x) \vee Q(y)) \vee R(x,y)))))$$

voidaan kirjoittaa selkeämmin

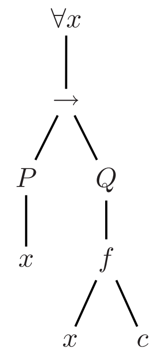
$$\exists x \forall y(\exists z(P(x,z) \wedge P(z,y)) \rightarrow Q(x) \vee Q(y) \vee R(x,y)).$$

Kaavojen jäsennyspuut

Predikaattilogiikan kaavoilla on yksikäsitteinen jäsennyspuu.

Kaavan $\forall x(P(x) \rightarrow Q(f(x,c)))$ jäsennyspuu on annettu oikealla.

Kyseinen kaava on muodoltaan *universaalisti kvantifioitu kaava*.



Huomio. Jäsennyspuun juuressa oleva konnektiivi määrää edelleen, mitä muotoa lause on. Esimerkiksi $\exists x P(x) \rightarrow \forall x P(x)$ on muodoltaan *implikaatio*, kun taas $\exists x(P(x) \rightarrow \forall y Q(y))$ on muodoltaan *eksistentiaalisesti kvantifioitu kaava*.

1.4 Kaavojen muodostaminen

1. Tunnistetaan kuvattavaan järjestelmään liittyvät objektit:
   - Otetaan käyttöön vakiosymboli jokaiselle objektille, johon on tarve viitata erikseen, eli *nimetään* tarvittavat objektit.
   - Mikäli objektien välillä on funktionaalisia riippuvuuksia, otetaan käyttöön vastaavat funktiosymbolit.
2. Tutkitaan millaisia relaatioita objektien välillä on ja otetaan käyttöön näitä vastaavat predikaattisymbolit.
3. Kuvataan relaatioiden väliset riippuvuudet kirjoittamalla niille määritelmät predikaattilogiikan kaavoina.

Huomio. Funktiosymbolin voi tarvittaessa korvata predikaattisymbolilla, mutta tällöin tällöin määritelmään tulee liittää funktionaalisuusehto.

Esimerkki.

Olkoon $t =$ "tuoli", $h =$ "hattu", $s =$ "sateenvarjo" vakioita ja $P(x, y) =$ "x on y:n päällä" kaksipaikkainen predikaatti. Tällöin:

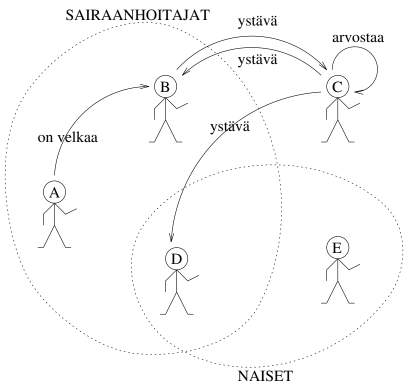
$\neg P(s, t) =$ "sateenvarjo ei ole tuolin päällä".

$\exists x P(x, h) =$
"on olemassa x, joka on hatun päällä",
eli "hatun päällä on jotakin".

$\exists x \forall y \neg P(y, x) =$
"on olemassa x siten, että mikään y ei ole x:n päällä",
eli "jonkin päällä ei ole mitään".

$\forall x (P(x, h) \rightarrow P(x, t)) =$
"kaikille x: jos x on hatun päällä, niin x on tuolin päällä",
eli "kaikki hatun päällä olevat ovat tuolin päällä".

Esimerkki. Kuvataan henkilöiden välisiä suhteita predikaattilogiikalla.



$N(e) \rightarrow A(c, c)$

$\exists x \exists y (Y(x, y) \wedge Y(y, x))$

$\exists x \exists y (S(x) \wedge S(y) \wedge V(x, y))$

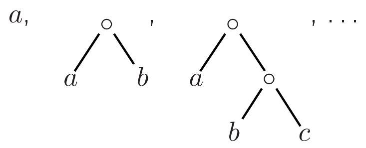
$\exists x A(x, x) \wedge \exists x (S(x) \wedge N(x))$

$\neg \forall x (S(x) \rightarrow N(x))$

$\forall x (Y(x, c) \rightarrow V(a, x))$

Esimerkki. Funktiosymbolit tarjoavat tavan esittää induktiivisia tietorakenteita termien avulla.

1. Luonnolliset luvut: vakiosymboli 0 ja funktiosymboli $s \in \mathcal{F}_1$.
   - Termit $0, s(0), s(s(0)), \ldots$ vastaavat luonnollisia lukuja $0, 1, 2, \ldots$
2. Listat: vakiosymboli e (tyhjä lista) ja funktiosymboli $c \in \mathcal{F}_2$.
   - Termit $e, c(a, e), c(a, c(b, e)), \ldots$ vastaavat listoja $[\,], [a], [a, b], \ldots$
3. Binääripuut: funktiosymbolit $l \in \mathcal{F}_1$ (lehtisolmut) ja $t \in \mathcal{F}_2$ (sisäsolmut).
   - Termit $l(a), t(l(a), l(b)), t(l(a), t(l(b), l(c))), \ldots$ vastaavat puita



Esimerkki. Esitetään binääripuut kuten edellä funktiosymbolien l ja t avulla. Kirjoitetaan määritelmä seuraavalle predikaatille:

$$P(x, y) = \text{“binääripuu } x \text{ on binääripuun } y \text{ peilikuva”.}$$

Koska binääripuut muodostavat induktiivisen tietorakenteen, on luontevaa, että predikaatille $P(x, y)$ joudutaan kirjoittamaan induktiivinen/rekursiivinen määritelmä seuraavalla tavalla.

- Perustapaus (pelkästä lehtisolmusta koostuvat binääripuut):
$$\forall x (P(l(x), l(x))).$$
- Induktioaskel (monimutkaisemmat binääripuut):
$$\forall x \forall y \forall z \forall v (P(x, y) \wedge P(z, v) \rightarrow P(t(x, z), t(v, y))).$$

$\Longrightarrow$ Määritelmä kattaa kaikki binääripuut.

1.5 Kvanttoreihin liittyviä määritelmiä

Kaavan *alikaavat* määräytyvät seuraavasti:

- Atomisen kaavan ψ ainoa alikaava on ψ itse.
- Kaavan $\exists x\psi$ ($\forall x\psi$) alikaavat ovat $\exists x\psi$ ($\forall x\psi$) ja ψ:n alikaavat.
- Lauselogiikan konnektiivit ($\neg, \rightarrow, \leftrightarrow, \lor, \land$) käsitellään vastaavalla tavalla (vrt. alilauseiden määritelmä lauselogiikan tapauksessa).

Esimerkki. Kaavan $\phi = \exists x A(x,x) \land \exists x(S(x) \land N(x))$ alikaavoja ovat

ϕ, $\exists x A(x,x)$, $\exists x(S(x) \land N(x))$, $A(x,x)$, $S(x) \land N(x)$, $S(x)$ ja $N(x)$.

Vapaat ja sidotut muuttujat kaavassa

Määritelmä. Olkoot $\exists x\psi$ ja $\forall x\phi$ predikaattilogiikan kaavoja. Alikaava ψ on kvanttorin $\exists x$ *vaikutusalue* kaavassa $\exists x\psi$. Vastaavasti alikaava ϕ on kvanttorin $\forall x$ vaikutusalue kaavassa $\forall x\phi$.

Esimerkki.

$$\forall x\, \overbrace{(P(x) \rightarrow \exists y\, \underbrace{Q(x,y)}_{\exists y})}^{\forall x}$$

$$\exists x\, \overbrace{(Q(x) \leftrightarrow \forall x\, \underbrace{P(x,z)}_{\forall x})}^{\exists x} \lor \exists x\, \overbrace{R(x)}^{\exists x}$$

Määritelmä. Muuttujan x esiintymä kaavassa ϕ on *sidottu*, jos se sijaitsee kvanttorin $\forall x$ (tai $\exists x$) vaikutusalueessa. Kvanttorisymbolia seuraava muuttujaesiintymä on sidottu.

Jos muuttujan x esiintymä ei ole sidottu, niin x:n esiintymä on *vapaa*.

Muuttuja x *esiintyy vapaana* ϕ:ssä, jos sillä on vapaa esiintymä ϕ:ssä.

Esimerkki. Tarkastellaan muuttujaesiintymiä seuraavassa lauseessa:

$$\forall \overbrace{x}^{\text{sid.}} (P(\overbrace{x}^{\text{sid.}}, \overbrace{y}^{\text{vap.}}, \overbrace{z}^{\text{vap.}}) \lor \exists \overbrace{y}^{\text{sid.}} (Q(\overbrace{y}^{\text{sid.}}) \rightarrow R(\overbrace{x}^{\text{sid.}}, \overbrace{z}^{\text{vap.}})))$$

Esimerkki. Kaavan $\exists x(P(x) \land \exists x Q(x))$ muuttujaesiintymät ovat sidotut.

Määritelmä. Kaavaa ϕ kutsutaan *lauseeksi*, jos siinä ei ole vapaita muuttujaesiintymiä.

Huomio. Luvussa 1.4 esitetyt kaavat ovat kaikki lauseita (tarkista).

Muuttujattoman termin sijoittaminen kaavaan

Määritelmä. Olkoon $\phi(x)$ kaava, jossa muuttuja x mahdollisesti esiintyy vapaana ja t muuttujaton termi. Kaavalla $\phi(t)$ tarkoitetaan kaavaa $\phi(x)$, jossa jokainen muuttujan x vapaa esiintymä on korvattu termillä t.

Esimerkki.

1. Olkoon $\phi(y) = \exists x(P(x,y) \lor Q(y,x))$.
   - Sijoittamalla muuttujattomat termit c ja $f(f(d))$ kaavaan $\phi(y)$ saadaan $\phi(c) = \exists x(P(x,c) \lor Q(c,x))$ ja

$$\phi(f(f(d))) = \exists x(P(x, f(f(d))) \lor Q(f(f(d)), x)).$$

2. Olkoon $\psi(x) = \exists x P(x) \land Q(x)$.
   - Sijoittamalla vakio c saadaan $\psi(c) = \exists x P(x) \land Q(c)$.

2 Predikaattilogiikan semantiikka

- Struktuurit
- Predikaattilogiikan totuusmääritelmä
- Semanttiset peruskäsitteet
- Vastamallit
- Peruskäsitteiden väliset yhteydet

2.1 Struktuurit

Predikaattilogiikassa totuusjakelut korvataan struktuureilla.

Määritelmä. *Struktuuri* (rakenne) $\mathcal{S}$ kielelle $\mathcal{L}$ koostuu

- universumista U, joka on jokin *ei-tyhjä* joukko, sekä
- vakio-, muuttuja-, funktio- ja predikaattisymbolien tulkinnoista:
  1. Vakiosymbolin $c \in \mathcal{C}$ tulkintana on alkio $c^{\mathcal{S}} \in U$.
  2. Muuttujasymbolin $v \in \mathcal{V}$ tulkintana on alkio $v^{\mathcal{S}} \in U$.
  3. Funktiosymbolin $f \in \mathcal{F}_n$ tulkintana on funktio $f^{\mathcal{S}}\colon U^n \to U$.
  4. Predikaattisymbolin $P \in \mathcal{P}_n$ tulkintana on relaatio $P^{\mathcal{S}} \subseteq U^n$.

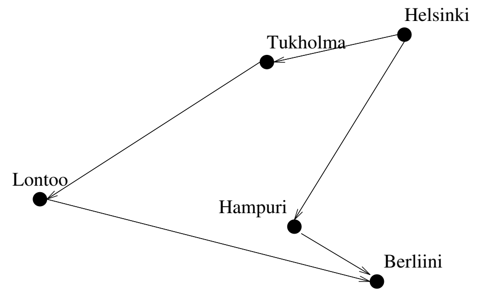
Struktuuri voidaan edelleen ymmärtää yhden asiaintilan kuvauksena.

Huomioita.

- Joukon $U^n = \overbrace{U \times \cdots \times U}^{n \text{ kpl}}$ alkiot ovat monikkoja (tai jonoja) $\langle a_1, \ldots, a_n \rangle$, missä alkiot $a_1 \in U$, ..., $a_n \in U$.
- Erikoistapaukset: $U^1 = U$ ja $U^0 = \{\langle\,\rangle\}$, missä $\langle\,\rangle$ on *tyhjä jono*.
- Kvanttoreilla $\exists v$ ja $\forall v$ tullaan jatkossa viittaamaan universumin eri alkioihin. Muuttujan v arvon vaihtaminen struktuurissa $\mathcal{S}$ tapahtuu seuraavalla tavalla.

Määritelmä. Struktuurilla $\mathcal{S}[v \mapsto a]$ tarkoitetaan struktuuria $\mathcal{S}'$, joka on muuten sama kuin $\mathcal{S}$, mutta muuttujasymbolin $v \in \mathcal{V}$ tulkintana $v^{\mathcal{S}'}$ onkin annettu alkio $a \in U$ (alkion $v^{\mathcal{S}}$ asemesta).

Esimerkki.



$$
\begin{aligned}
U &= \{he, tu, ha, be, lo\} & \text{Helsinki}^{\mathcal{S}} &= he\\
\text{Tukholma}^{\mathcal{S}} &= tu & \text{Hampuri}^{\mathcal{S}} &= ha\\
\text{Berliini}^{\mathcal{S}} &= be & \text{Lontoo}^{\mathcal{S}} &= lo\\
\text{pääkaupunki}^{\mathcal{S}} &= \{he, tu, be, lo\} \subseteq U^1 = U\\
\text{lento}^{\mathcal{S}} &= \{\langle he, tu\rangle, \langle tu, lo\rangle, \langle lo, be\rangle, \langle he, ha\rangle, \langle ha, be\rangle\} \subseteq U^2
\end{aligned}
$$

2.2 Predikaattilogiikan totuusmääritelmä

Termien tulkinta struktuurissa

Määritelmä. Olkoon $\mathcal{S}$ struktuuri kielelle $\mathcal{L}$ ja U struktuurin $\mathcal{S}$ universumi.

- Vakio $c \in \mathcal{C}$ *nimeää* universumin U alkion $c^{\mathcal{S}}$.
- Muuttuja $v \in \mathcal{V}$ *nimeää* universumin U alkion $v^{\mathcal{S}}$.
- Jos termit $t_1, \ldots, t_n$ nimeävät universumin U alkiot $t_1^{\mathcal{S}}, \ldots, t_n^{\mathcal{S}}$ ja $f \in \mathcal{F}_n$, niin termi $f(t_1, \ldots, t_n)$ *nimeää* universumin U alkion $f^{\mathcal{S}}(t_1^{\mathcal{S}}, \ldots, t_n^{\mathcal{S}})$.

Näin voimme viitata kielen $\mathcal{L}$ termeillä universumin U alkioihin (kunhan vakio-, muuttuja-, ja funktiosymbolien tulkinnat on annettu).

Esimerkki. Tarkastellaan vakiosymbolia $c \in \mathcal{C}$ ja funktiosymboleja $f \in \mathcal{F}_1$ ja $g \in \mathcal{F}_2$. Olkoon struktuurin $\mathcal{S}$ universumina U luonnollisten lukujen joukko $0, 1, 2, \ldots$. Valitaan em. symbolien tulkinnat seuraavasti:

$$\begin{aligned} c^{\mathcal{S}} &= 0, \\ f^{\mathcal{S}} &= \text{seuraajafunktio, eli } f^{\mathcal{S}}(n) = n+1, \text{ ja} \\ g^{\mathcal{S}} &= \text{summafunktio, eli } g^{\mathcal{S}}(n,m) = n+m. \end{aligned}$$

Tällöin c nimeää alkion 0,
$f(c)$ nimeää alkion 1,
$f^n(c) = \underbrace{f(f(\ldots f}_{n \text{ kpl}}(c)\ldots))$ nimeää alkion n ja
$g(f(c), f(f(c)))$ nimeää alkion 3.

Kaavojen totuusarvojen laskeminen struktuurissa

Olkoon $\mathcal{S}$ struktuuri kielelle $\mathcal{L}$ ja U struktuurin $\mathcal{S}$ universumi.

Määritelmä. Seuraavassa määritellään, milloin kaava $\phi \in \mathcal{L}$ on *tosi* struktuurissa $\mathcal{S}$ (merk. $\mathcal{S} \models \phi$) ja milloin *epätosi* (merk. $\mathcal{S} \not\models \phi$).

1. $\mathcal{S} \models t_1 = t_2 \iff t_1^{\mathcal{S}}$ ja $t_2^{\mathcal{S}}$ ovat sama universumin U alkio (yllä t_1 ja t_2 ovat mitä tahansa termejä).
2. $\mathcal{S} \models P(t_1, \ldots, t_n) \iff$
   $\langle t_1, \ldots, t_n \rangle^{\mathcal{S}}$ (eli jono $\langle t_1^{\mathcal{S}}, \ldots, t_n^{\mathcal{S}} \rangle$) kuuluu tulkintaan $P^{\mathcal{S}}$
   (yllä $n > 0$, $P \in \mathcal{P}_n$ ja $t_1, \ldots, t_n$ ovat mitä tahansa termejä).
3. $\mathcal{S} \models P \iff$ tyhjä jono $\langle\rangle$ kuuluu tulkintaan $P^{\mathcal{S}}$ (missä $P \in \mathcal{P}_0$).
4. $\mathcal{S} \models \neg\alpha \iff \mathcal{S} \not\models \alpha$.

5. $\mathcal{S} \models \alpha \wedge \beta \iff \mathcal{S} \models \alpha$ ja $\mathcal{S} \models \beta$.
6. $\mathcal{S} \models \alpha \vee \beta \iff \mathcal{S} \models \alpha$ tai $\mathcal{S} \models \beta$.
7. $\mathcal{S} \models \alpha \rightarrow \beta \iff \mathcal{S} \not\models \alpha$ tai $\mathcal{S} \models \beta$.
8. $\mathcal{S} \models \alpha \leftrightarrow \beta \iff$ joko $\mathcal{S} \models \alpha$ ja $\mathcal{S} \models \beta$, tai $\mathcal{S} \not\models \alpha$ ja $\mathcal{S} \not\models \beta$.
9. $\mathcal{S} \models \exists x\ \alpha(x) \iff$
   $\mathcal{S}[x \mapsto a] \models \alpha(x)$ *jollekin* universumin U alkiolle $a \in U$.
10. $\mathcal{S} \models \forall x\ \alpha(x) \iff$
    $\mathcal{S}[x \mapsto a] \models \alpha(x)$ *kaikille* universumin U alkioille $a \in U$.

Väite. Kaikille kaavoille $\phi \in \mathcal{L}$ pätee joko $\mathcal{S} \models \phi$ tai $\mathcal{S} \not\models \phi$.

Väite. Jos kaava $\phi \in \mathcal{L}$ on lisäksi *lause*, sen totuusarvo ei riipu muuttujien $v \in \mathcal{V}$ tulkinnoista struktuurissa $\mathcal{S}$.

Esimerkki. Tarkastellaan graafin solmujen joukkoa (universumi) ja esitetään kaaret kaksipaikkaisen predikaatin K avulla. Nyt esim. graafia

$$\begin{tikzcd} s_0 \arrow[r] & s_1 \arrow[r] & s_2 \arrow[loop right] \end{tikzcd}$$

vastaa struktuuri $\mathcal{S}$, jonka universumi on $U = \{s_0, s_1, s_2\}$ ja K:n tulkinta $K^{\mathcal{S}} = \{\langle s_0, s_1\rangle, \langle s_1, s_2\rangle, \langle s_2, s_2\rangle\}$ (x ja y ovat tulkittavissa vapaasti).

1. $\langle s_0, s_1\rangle \in K^{\mathcal{S}} \implies \langle x, y\rangle^{\mathcal{S}[x\mapsto s_0, y\mapsto s_1]} \in K^{\mathcal{S}[x\mapsto s_0, y\mapsto s_1]}$
   $\implies \mathcal{S}[x \mapsto s_0, y \mapsto s_1] \models K(x, y)$
   $\implies \mathcal{S}[x \mapsto s_0] \models \exists y K(x, y).$
2. Vastaavasti $\langle s_1, s_2\rangle \in K^{\mathcal{S}} \implies \mathcal{S}[x \mapsto s_1] \models \exists y K(x, y).$
3. Vastaavasti $\langle s_2, s_2\rangle \in K^{\mathcal{S}} \implies \mathcal{S}[x \mapsto s_2] \models \exists y K(x, y).$

Esimerkki. (jatkoa)

4. Koska $\mathcal{S}[x \mapsto s_i] \models \exists y K(x, y)$ jokaiselle universumin alkiolle $s_i \in U$, voimme todeta, että $\mathcal{S} \models \forall x \exists y K(x, y)$.
5. Lisäksi esim.

$$\begin{aligned} \langle s_2, s_2\rangle \in K^{\mathcal{S}} &\implies \langle x, x\rangle^{\mathcal{S}[x\mapsto s_2]} \in K^{\mathcal{S}[x\mapsto s_2]} \\ &\implies \mathcal{S}[x \mapsto s_2] \models K(x, x) \\ &\implies \mathcal{S}[x \mapsto s_2] \not\models \neg K(x, x) \\ &\implies \mathcal{S} \not\models \forall x \neg K(x, x) \\ &\implies \mathcal{S} \models \neg\forall x \neg K(x, x). \end{aligned}$$

Mieti millaisia graafin ominaisuuksia lauseet $\forall x \exists y K(x, y)$ ja $\neg\forall x \neg K(x, x)$ itse asiassa tarkoittavat!

Esimerkki. Tarkastellaan seuraavia opiskelijatietokannan relaatiotauluja:

Op (opiskelija)

3310D	Taina
4820H	Otso
9055Z	Elsi
4358E	Uljas
...	...

Ko (koulutusohjelma)

4358E	Tieto
3310D	Sähkö
9055Z	Kemia
4820H	Rakennus
...	...

1. *Kyselyn* $\exists x(\mathrm{Op}(x, \mathrm{Elsi}) \wedge \mathrm{Ko}(x, \mathrm{Tieto}))$ *evaluointi* tarkoittaa kyseisen kaavan totuusarvon laskentaa vastaavassa struktuurissa.
2. Kyselyyn $\neg\forall x(\mathrm{Op}(x, \mathrm{Elsi}) \rightarrow \neg\mathrm{Ko}(x, \mathrm{Tieto}))$ saadaan sama vastaus. Miten tämä on perusteltavissa?

2.3 Semanttiset peruskäsitteet

- Semanttisten peruskäsitteiden määritelmät ovat muodoltaan samat.
- Olennaisena erona lauselogiikkaan on, että lauseiden rakenne on monipuolisempi ja että struktuurit korvaavat totuusjakelut.

Määritelmä. Struktuuri $\mathcal{S}$ on lauseen $\alpha \in \mathcal{L}$ *malli*, joss lause α on tosi struktuurissa $\mathcal{S}$ eli $\mathcal{S} \models \alpha$.

Määritelmä. Struktuuri $\mathcal{S}$ on lausejoukon $\Sigma \subseteq \mathcal{L}$ malli, joss kaikille lausejoukon Σ lauseille $\sigma \in \Sigma$ pätee $\mathcal{S} \models \sigma$.

Määritelmä. Lause $\alpha \in \mathcal{L}$ (tai lausejoukko $\Sigma \subseteq \mathcal{L}$) on *toteutuva*, joss ainakin yksi struktuuri $\mathcal{S}$ on sen malli.

Esimerkki. $\exists x \forall y P(x, y)$ on toteutuva.
Olkoon struktuurin $\mathcal{S}$ universumi $U = \{1, 2\}$ ja $P^{\mathcal{S}} = \{\langle 1, 1\rangle, \langle 1, 2\rangle\}$.

1. $\langle 1,1\rangle \in P^{\mathcal{S}} \implies \langle x, y\rangle^{\mathcal{S}[x\mapsto 1, y\mapsto 1]} \in P^{\mathcal{S}[x\mapsto 1, y\mapsto 1]}$
   $\implies \mathcal{S}[x \mapsto 1, y \mapsto 1] \models P(x, y).$
2. $\langle 1,2\rangle \in P^{\mathcal{S}} \implies \langle x, y\rangle^{\mathcal{S}[x\mapsto 1, y\mapsto 2]} \in P^{\mathcal{S}[x\mapsto 1, y\mapsto 2]}$
   $\implies \mathcal{S}[x \mapsto 1, y \mapsto 2] \models P(x, y).$
3. Siis $\mathcal{S}[x \mapsto 1] \models \forall y P(x, y)$ ja $\mathcal{S} \models \exists x \forall y P(x, y)$.

Määritelmä. Kielen $\mathcal{L}$ lause α on *pätevä* (merkitään $\models \alpha$), joss $\mathcal{S} \models \alpha$ kaikissa $\mathcal{L}$:n struktuureissa $\mathcal{S}$.

Esimerkki. Osoitetaan $\models \forall x P(x) \leftrightarrow \neg\exists x \neg P(x)$.

Kaikille kielen $\mathcal{L}$ struktuureille $\mathcal{S}$ ja vastaaville universumeille U pätee:

$$\begin{array}{ll}
 & \mathcal{S} \models \forall x P(x) \\
\iff & \mathcal{S}[x \mapsto a] \models P(x) \text{ kaikille } a \in U \\
\iff & \text{ei ole niin, että } \mathcal{S}[x \mapsto a] \not\models P(x) \text{ jollekin } a \in U \\
\iff & \text{ei ole niin, että } \mathcal{S}[x \mapsto a] \models \neg P(x) \text{ jollekin } a \in U \\
\iff & \mathcal{S} \not\models \exists x \neg P(x) \\
\iff & \mathcal{S} \models \neg\exists x \neg P(x).
\end{array}$$

Siis $\mathcal{S} \models \forall x P(x) \leftrightarrow \neg\exists x \neg P(x)$ struktuurista $\mathcal{S}$ riippumatta.

Esimerkki. $\models \forall x P(x) \lor \neg\forall x P(x)$
Olkoon $\mathcal{S}$ mielivaltainen $\mathcal{L}$:n struktuuri.
Nyt $\mathcal{S} \models \forall x P(x) \iff \mathcal{S} \not\models \neg\forall x P(x)$, joten $\mathcal{S} \models \forall x P(x) \lor \neg\forall x P(x)$.

Määritelmä. Kielen $\mathcal{L}$ lause α on lausejoukon $\Sigma \subseteq \mathcal{L}$ *looginen seuraus* (merkitään $\Sigma \models \alpha$), joss α on tosi jokaisessa lausejoukon Σ mallissa $\mathcal{S}$.

Esimerkki. $\{\forall x P(x)\} \models \exists x P(x)$

Olkoon $\mathcal{S}$ struktuuri siten, että $\mathcal{S} \models \forall x P(x)$. Tällöin

$$\begin{array}{ll}
\implies & \text{kaikille } a \in U \text{ pätee } \mathcal{S}[x \mapsto a] \models P(x) \\
\implies & \text{jollekin } a \in U \text{ pätee } \mathcal{S}[x \mapsto a] \models P(x) \\
 & \text{(universumi } U \text{ on ei-tyhjä struktuurin määritelmän perusteella)} \\
\implies & \mathcal{S} \models \exists x P(x).
\end{array}$$

2.4 Vastamallit

Vastamallin käsitettä käytetään myös predikaattilogiikan tapauksessa.
Olkoon $\phi \in \mathcal{L}$ mikä tahansa lause ja $\Sigma \subseteq \mathcal{L}$ mikä tahansa lausejoukko.

- Jos $\not\models \phi$, niin vastamalli on struktuuri $\mathcal{S}$ siten, että $\mathcal{S} \not\models \phi$.
- Jos $\Sigma \not\models \phi$, niin vastamalli on malli $\mathcal{S} \models \Sigma$ siten, että $\mathcal{S} \not\models \phi$.

Esimerkki.
$\{\forall x(A(x) \rightarrow B(x)), \forall y A(y) \lor \forall y \neg A(y)\} \not\models \forall z B(z) \lor \forall z \neg B(z)$, koska voimme muodostaa esim. seuraavan vastamallin (struktuurin) $\mathcal{S}$:
$U = \{1, 2\}$, $A^{\mathcal{S}} = \emptyset$, ja $B^{\mathcal{S}} = \{1\}$.

Täten $\mathcal{S} \models \forall x(A(x) \rightarrow B(x))$, $\mathcal{S} \models \forall y \neg A(y)$, $\mathcal{S} \models \forall y A(y) \lor \forall y \neg A(y)$, $\mathcal{S} \not\models \forall z B(z)$, $\mathcal{S} \not\models \forall z \neg B(z)$ ja edelleen $\mathcal{S} \not\models \forall z B(z) \lor \forall z \neg B(z)$.

(Muuttujien x, y ja z tulkinnat voidaan valita vapaasti struktuurissa $\mathcal{S}$).

Esimerkki. Luokitellaan viikonpäiviä seuraavalla lausejoukolla:

$$\Sigma = \left\{ \begin{array}{l} \mathsf{tentti}(\mathsf{tiistai}), \\ \mathsf{tentti}(\mathsf{keskiviikko}), \\ \mathsf{luento}(\mathsf{keskiviikko}), \\ \forall x(\neg \mathsf{tentti}(x) \land \neg \mathsf{luento}(x) \rightarrow \mathsf{vapaa}(x)) \end{array} \right\}.$$

Onko $\Sigma \models \mathsf{vapaa}(\mathsf{perjantai})$? *Ei*, koska löytyy *vastamalli* $\mathcal{S}$, jonka perusteella $\Sigma \not\models \mathsf{vapaa}(\mathsf{perjantai})$ eli $\mathcal{S} \models \Sigma$ ja $\mathcal{S} \not\models \mathsf{vapaa}(\mathsf{perjantai})$!

Olkoon universumi $U = \{t, k, p\}$ ja symbolien tulkinnat seuraavat:

$$\begin{array}{lcllcl} \mathsf{tiistai}^{\mathcal{S}} & = & t, & \mathsf{keskiviikko}^{\mathcal{S}} & = & k, \\ \mathsf{perjantai}^{\mathcal{S}} & = & p, & \mathsf{tentti}^{\mathcal{S}} & = & \{t, k\}, \\ \mathsf{luento}^{\mathcal{S}} & = & \{k, p\}, & \mathsf{vapaa}^{\mathcal{S}} & = & \{\}. \end{array}$$

Mutta: $\Sigma \cup \{\neg \mathsf{tentti}(\mathsf{perjantai}), \neg \mathsf{luento}(\mathsf{perjantai})\} \models \mathsf{vapaa}(\mathsf{perjantai})$.

2.5 Peruskäsitteiden väliset yhteydet

Seuraavat ominaisuudet ovat voimassa myös predikaattilogiikassa:

- $\models \alpha \iff \neg\alpha$ on toteutumaton.
- $\Sigma \models \alpha \iff \Sigma \cup \{\neg\alpha\}$ on toteutumaton.
- $\models \alpha \iff \emptyset \models \alpha$.
- $\{\alpha_1, \ldots, \alpha_n\} \models \alpha \iff \models \alpha_1 \land \cdots \land \alpha_n \rightarrow \alpha$.

Olkoon $\mathsf{Cn}(\Sigma) = \{\phi \in \mathcal{L} \mid \phi$ on lause ja $\Sigma \models \phi\}$ lausejoukolle $\Sigma \subseteq \mathcal{L}$:

- $\Sigma \subseteq \mathsf{Cn}(\Sigma)$ ja $\Sigma \equiv \mathsf{Cn}(\Sigma)$.
- Monotonisuus: $\Sigma_1 \subseteq \Sigma_2 \implies \mathsf{Cn}(\Sigma_1) \subseteq \mathsf{Cn}(\Sigma_2)$.
- $\mathsf{Cn}(\mathsf{Cn}(\Sigma)) = \mathsf{Cn}(\Sigma)$.

3 Normaalimuodot

- Prenex-normaalimuoto
- Konjunktiivinen normaalimuoto
- Eksistenssikvanttorien eliminointi
- Lauseiden klausuulimuoto predikaattilogiikassa

3.1 Prenex-normaalimuoto

Lause α on *prenex*-normaalimuodossa, mikäli se on muotoa

$$\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \cdots \mathcal{Q}_n x_n \phi,$$

missä jokainen $\mathcal{Q}_i$ on jompikumpi kvanttoreista ($\forall$ tai $\exists$) ja alikaava ϕ ei sisällä kvanttoreita.

Esimerkki. Seuraavat lauseet ovat prenex-normaalimuodossa:
$P(a)$, $\forall x P(x)$, $\forall x \exists y\, P(x, y)$ ja
$\forall x \exists y \forall z \forall w (P(x, y, z) \rightarrow (Q(y, z, w) \rightarrow R(z, w, x)))$.

Väite. Jokainen predikaattilogiikan lause on loogisesti ekvivalentti jonkin prenex-normaalimuodossa olevan lauseen kanssa.

Lauseiden muuttaminen prenex-normaalimuotoon

Mikä tahansa predikaattilogiikan lause voidaan muuttaa prenex-normaalimuotoon seuraavalla menettelyllä:

1. Poistetaan konnektiivit $\rightarrow$ ja $\leftrightarrow$:

$$\phi \rightarrow \psi \rightsquigarrow \neg\phi \vee \psi$$
$$\phi \leftrightarrow \psi \rightsquigarrow (\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$$

2. Viedään negaatiot kaavarakenteen sisään (atomisten kaavojen eteen) seuraavien muunnosten avulla:

$$\neg\neg\varphi \rightsquigarrow \varphi$$
$$\neg(\varphi \wedge \psi) \rightsquigarrow \neg\varphi \vee \neg\psi \qquad \neg\forall y\varphi \rightsquigarrow \exists y\neg\varphi$$
$$\neg(\varphi \vee \psi) \rightsquigarrow \neg\varphi \wedge \neg\psi \qquad \neg\exists y\varphi \rightsquigarrow \forall y\neg\varphi$$

3. Tuodaan kvanttorit ulos kaavarakenteesta:

$$\forall y\varphi(y) \vee \psi \rightsquigarrow \forall z(\varphi(z) \vee \psi)$$
$$\forall y\varphi(y) \wedge \psi \rightsquigarrow \forall z(\varphi(z) \wedge \psi)$$
$$\psi \vee \forall y\varphi(y) \rightsquigarrow \forall z(\psi \vee \varphi(z))$$
$$\psi \wedge \forall y\varphi(y) \rightsquigarrow \forall z(\psi \wedge \varphi(z))$$

- Yllä y korvataan uudella muuttujalla z, mikäli y esiintyy vapaana alikaavassa ψ. Muussa tapauksessa z voi aivan hyvin olla y.
- Eksistenssikvanttorit $\exists y$ käsitellään samaan tapaan (saadaan 4 vastaavanmuotoista sääntöä lisää).

Huomio. Muuttujan korvaaminen uudella on olennaista:

$$\forall xP(x) \vee \forall xQ(x) \rightsquigarrow \forall x(P(x) \vee \forall xQ(x)) \rightsquigarrow \forall x\forall y(P(x) \vee Q(y)).$$

(Lauseet $\forall xP(x) \vee \forall xQ(x)$ ja $\forall x(P(x) \vee Q(x))$ eivät ole ekvivalentit!)

Esimerkki. Muunnetaan seuraava lause prenex-normaalimuotoon:

$$\begin{aligned}
& \forall x(P(x) \rightarrow \exists zR(z,x)) \rightarrow \exists xQ(x) \\
\rightsquigarrow\ & \neg\forall x(P(x) \rightarrow \exists zR(z,x)) \vee \exists xQ(x) \\
\rightsquigarrow\ & \neg\forall x(\neg P(x) \vee \exists zR(z,x)) \vee \exists xQ(x) \\
\rightsquigarrow\ & \exists x\neg(\neg P(x) \vee \exists zR(z,x)) \vee \exists xQ(x) \\
\rightsquigarrow\ & \exists x(\neg\neg P(x) \wedge \neg\exists zR(z,x)) \vee \exists xQ(x) \\
\rightsquigarrow\ & \exists x(P(x) \wedge \forall z\neg R(z,x)) \vee \exists xQ(x) \\
\rightsquigarrow\ & \exists x(\exists x(P(x) \wedge \forall z\neg R(z,x)) \vee Q(x)) \\
\rightsquigarrow\ & \exists x\exists y((P(y) \wedge \forall z\neg R(z,y)) \vee Q(x)) \\
\rightsquigarrow\ & \exists x\exists y(\forall z(P(y) \wedge \neg R(z,y)) \vee Q(x)) \\
\rightsquigarrow\ & \exists x\exists y\forall z((P(y) \wedge \neg R(z,y)) \vee Q(x)).
\end{aligned}$$

3.2 Konjunktiivinen normaalimuoto

Määritelmä. *Literaalit* ovat joko

1. atomikaavoja $P(\vec{t})$ (eli *positiivisia literaaleja*) tai
2. atomikaavojen negaatioita $\neg P(\vec{t})$ (eli *negatiivisia literaaleja*).

Yllä $\vec{t}$ tarkoittaa termien $t_1, \ldots, t_n$ sekvenssiä kun $P \in \mathcal{P}_n$. Jos $P \in \mathcal{P}_0$, sekvenssi on tyhjä ja tällöin myös sulut jätetään kirjoittamatta näkyviin.

Määritelmä. Lause α on konjunktiivisessa normaalimuodossa, mikäli se on on prenex-normaalimuodossa $\mathcal{Q}_1x_1\mathcal{Q}_2x_2\cdots\mathcal{Q}_nx_n\phi$, missä kvanttoreita sisältämätön osa $\phi = \phi_1 \wedge \cdots \wedge \phi_n$ ja jokainen konjunktion jäsen ϕ_i on muodoltaan literaalien disjunktio.

Esimerkki. Seuraava lause on konjunktiivisessa normaalimuodossa: $\forall x\exists y\forall z((\neg P(x,y) \vee Q(y,x)) \wedge R(z) \wedge (\neg R(x) \vee P(y,z) \vee Q(x,z)))$.

- Tarvittaessa prenex-normaalimuodon kvanttoreita sisältämätön osa voidaan järjestää konjunktiiviseen normaalimuotoon seuraavasti:

$$\begin{aligned} \varphi \vee (\phi \wedge \psi) &\rightsquigarrow (\varphi \vee \phi) \wedge (\varphi \vee \psi) \\ (\phi \wedge \psi) \vee \varphi &\rightsquigarrow (\phi \vee \varphi) \wedge (\psi \vee \varphi) \end{aligned}$$

- Näin ollen voimme todeta seuraavan tuloksen:

Väite. Jokainen predikaattilogiikan lause on loogisesti ekvivalentti jonkin konjunktiivisessa normaalimuodossa olevan lauseen kanssa.

Esimerkki. Muunnetaan edellä johdettu prenex-normaalimuoto edelleen konjunktiiviseksi normaalimuodoksi:

$$\begin{aligned} & \exists x \exists y \forall z ((P(y) \wedge \neg R(z,y)) \vee Q(x)) \\ \rightsquigarrow\ & \exists x \exists y \forall z ((P(y) \vee Q(x)) \wedge (\neg R(z,y) \vee Q(x))). \end{aligned}$$

3.3 Eksistenssikvanttorien eliminointi

Esimerkki. Tarkastellaan kahta kokonaislukuja koskevaa väittämää:

1. Summafunktiolla on vasen identiteetti:

$$\exists x \forall y (x + y = y).$$

   Identiteettialkio voidaan nimetä vakiosymbolilla 0:

$$\forall y (0 + y = y).$$

2. Jokaisella kokonaisluvulla on vastaluku:

$$\forall x \exists y (x + y = 0).$$

   Vastalukufunktio voidaan nimetä funktiosymbolilla $-$:

$$\forall x (x + -(x) = 0).$$

Eksistenssikvanttorien eliminointi yleisessä tapauksessa

Olkoon $\vec{\mathcal{Q}}x\phi$ prenex-normaalimuodossa ja $\exists x$ kvanttorisekvenssin $\vec{\mathcal{Q}}x$ ensimmäinen eksistenssikvanttori.

1. Jos kvanttori $\exists x$ on sekvenssissä $\vec{\mathcal{Q}}x$ vieläpä ensimmäisenä, poistetaan $\exists x$ sekvenssistä ja korvataan näin syntyvät muuttujan x vapaat esintymät jollain uudella vakiolla c.
2. Jos kvanttoria $\exists x$ esiintyy sekvenssissä $\vec{\mathcal{Q}}x$ universaalikvanttorien $\forall y_1 \cdots \forall y_n$ jälkeen, poistetaan kvanttori $\exists x$ ja korvataan näin syntyvät muuttujan x vapaat esiintymät termillä $f(y_1, \ldots, y_n)$, missä f on uusi funktiosymboli.

Huomio. Jälkimmäisessä tapauksessa funktion f avulla kuvataan muuttujan x *mahdollinen* riippuvuus muuttujista $y_1, \ldots, y_n$.

- Eksistenssikvanttorien eliminointia kutsutaan kehittäjänsä mukaan Skolemoinniksi ja vastaavasti ko. prosessissa valittavia uusia vakio- ja funktiosymboleita *Skolem-vakioiksi ja -funktioiksi*.

Esimerkki. Suoritetaan Skolemointi seuraaville lauseille:

$$\begin{aligned} \exists x P(x) &\rightsquigarrow P(c) \\ \exists x \forall y \exists x P(x,y) &\rightsquigarrow \forall y P(f(y),y) \\ \forall x \exists y (P(x,y) \wedge Q(y,x)) &\rightsquigarrow \forall x\ (P(x,f(x)) \wedge Q(f(x),x)) \\ \forall x \exists y \forall z \exists w P(w,z,y,x) &\rightsquigarrow \forall x \forall z P(g(x,z),z,f(x),x) \\ \exists x \exists y \forall z P(x,y,z,x) &\rightsquigarrow \forall z P(c_1,c_2,z,c_1) \\ \forall x \exists y \exists z P(x,y,z) &\rightsquigarrow \forall x P(x,f_1(x),f_2(x)) \end{aligned}$$

Skolemoinnin loogiset ominaisuudet

Väite. Prenex-normaalimuodossa oleva lause φ on toteutuva $\iff$ lauseen φ skolemoitu muoto φ' on toteutuva.

Huomio. Prenex-normaalimuodossa olevan lauseen φ skolemoitu muoto φ' **ei välttämättä ole** loogisesti ekvivalentti lauseen φ kanssa.

Esimerkki. Lause $\exists x P(x)$ ja sen skolemoitu muoto $P(c)$.

Nyt $\models P(c) \rightarrow \exists x P(x)$, mutta $\not\models \exists x P(x) \rightarrow P(c)$.

Vastamalli $\mathcal{S}$: universumi $U = \{1, 2\}$, $c^{\mathcal{S}} = 1$ ja $P^{\mathcal{S}} = \{2\}$.
Nyt $\mathcal{S} \models \exists x P(x)$, mutta $\mathcal{S} \not\models P(c)$.

Täten myös $\not\models \exists x\ P(x) \leftrightarrow P(c)$ ja edelleen $\exists x P(x) \not\equiv P(c)$.

3.4 Lauseiden klausuulimuoto predikaattilogiikassa

Mille tahansa lauseelle voidaan hakea klausuulimuoto seuraavasti:

1. Haetaan prenex-normaalimuoto.
2. Muunnetaan tämä konjunktiiviseen normaalimuotoon.
3. Tarvittaessa poistetaan eksistenssikvanttorit Skolemoimalla.
4. Kirjoitetaan klausuuliesitys (joukko literaalien joukkoja).

Esimerkki. Klausuuliesitys lauseelle $\forall x(\neg(P(x) \rightarrow \forall y Q(x, y)) \vee R(x))$:

$$\begin{aligned}
&\rightsquigarrow \forall x \exists z((P(x) \wedge \neg Q(x, z)) \vee R(x)) && (1)\\
&\rightsquigarrow \forall x \exists z((P(x) \vee R(x)) \wedge (\neg Q(x, z) \vee R(x))) && (2)\\
&\rightsquigarrow \forall x((P(x) \vee R(x)) \wedge (\neg Q(x, f(x)) \vee R(x))) && (3)\\
&\rightsquigarrow \{\{P(x), R(x)\}, \{\neg Q(x, f(x)), R(x)\}\}. && (4)
\end{aligned}$$

Huomio. Jos lause on muodoltaan *konjunktio* $\phi_1 \wedge \cdots \wedge \phi_n$, on mahdollista saattaa konjunktion jäsenet $\phi_1, \ldots, \phi_n$ klausuulimuotoon *erikseen*. Muista myös, että $\neg(\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi) \equiv \phi_1 \wedge \cdots \wedge \phi_n \wedge \neg\phi$.

Esimerkki. $\forall x P(x) \wedge \exists x Q(x)$. Konjunktion jäsenille saadaan klausuuliesitykset $\{\{P(x)\}\}$ ja $\{\{Q(c)\}\}$ ja näinollen koko lauseen klausuuliesitykseksi näiden unioni $\{\{P(x)\}, \{Q(c)\}\}$.

Huomio. Existenssikvanttorit kannattaa tuoda ulos lauserakenteesta ennen universaalikvanttoreita (mikäli mahdollista).

Esimerkki. Siis $\forall x P(x) \vee \exists x Q(x)$ kirjoitetaan muotoon $\exists x \forall y(P(y) \vee Q(x))$ eikä muotoon $\forall x \exists y(P(x) \vee Q(y))$.

Näin saatetaan välttää Skolem-funktioiden käyttöönotto tai ainakin vähennetään Skolem-funktioiden argumenttien lukumäärää $\Longrightarrow$ klausuulimuodosta tulee rakenteeltaan yksinkertaisempi.

4 Semanttiset taulut predikaattilogiikalle

- Taulusäännöt kvanttoreiden käsittelyyn
- Semanttiset taulut predikaattilogiikalle
- Ohjeita taulutodistusten laadintaan
- Systemaattinen taulu
- Vastamallin konstruointi

4.1 Taulusäännöt kvanttorien käsittelyyn

- Muotoa $T\,\exists x\varphi(x)$ (tai $E\,\forall x\varphi(x)$) oleva solmu tulee hajoittaa *kertaalleen* käyttäen jotain (hajoitushetkellä) *uutta vakiota* c.

$T\,\exists x\varphi(x)$	$E\,\forall x\varphi(x)$
$\vert$	$\vert$
$T\varphi(c)$	$E\varphi(c)$
c uusi vakio	c uusi vakio

- Olkoon P polku (juurisolmusta lehtisolmuun), jolla solmu $T\,\exists x\varphi(x)$ ($E\,\forall x\varphi(x)$) esiintyy ja jota on tarkoitus jatkaa ao. taulusäännöllä. Vakio c on *uusi*, mikäli se ei esiinny polulla P *hajoittamishetkellä*.

Huomio. Uuden vakion käyttöönotto johtuu siitä, ettemme tiedä, millä universumin alkiolla on ko. ominaisuus φ (tai ei ole ominaisuutta φ).

- Muotoa $T\,\forall x\varphi(x)$ (tai $E\,\exists x\varphi(x)$) oleva solmu tulee (tarvittaessa) hajoittaa kaikille *muuttujattomille termeille* t (vakioita tai vakioista ja funktiosymboleista rakentuvia monimutkaisempia termejä).

$T\,\forall x\varphi(x)$	$E\,\exists x\varphi(x)$
$\vert$	$\vert$
$T\varphi(t)$	$E\varphi(t)$
t muuttujaton termi	t muuttujaton termi

Huomio.

- Muuttujattomia termejä voi olla ääretön määrä, joten muotoa $T\,\forall x\varphi(x)$ (tai $E\,\exists x\varphi(x)$) olevaa solmua ei välttämättä saada hajoitetuksi soveltamalla ao. taulusääntöä äärellisen monta kertaa.

Esimerkki. Juurisolmusta $T\,\exists x\forall y\exists z P(x,y,z)$ muodostettua semanttista taulua ei saada valmiiksi äärellisellä määrällä hajoituksia.

1. $T\,\exists x\forall y\exists z P(x,y,z)$
2. $T\,\forall y\exists z P(a,y,z)$ $^{1.\ x/a}$ (a uusi vakio)
3. $T\,\exists z P(a,a,z)$ $^{2.\ y/a}$
4. $TP(a,a,b)$ $^{3.\ z/b}$ (b uusi vakio)
5. $T\,\exists z P(a,b,z)$ $^{2.\ y/b}$
6. $TP(a,b,c)$ $^{5.\ z/c}$ (c uusi vakio)

jne.

4.2 Semanttiset taulut predikaattilogiikalle

- Semanttisten taulujen määritelmä säilyy ennallaan.
- Rajaamme yhtäsuuruuspredikaatin "$=$" tarkastelun ulkopuolelle välttääksemme yhtälöiden käsittelyn (vrt. *equality axioms*).
- Ehtoja, millä polun solmu on *hajoitettu*, joudutaan täydentämään:

  Olkoon τ semanttinen taulu ja P polku juurisolmusta lehtisolmuun τ:ssa. P:n solmu $T\,\forall x\varphi(x)$ ($E\,\exists x\varphi(x)$) hajoitettu polulla P, jos

  – polulla P esiintyy $T\varphi(t)$ ($E\varphi(t)$) kaikille muuttujattomille termeille t, jotka voidaan muodostaa polulla P esiintyvistä vakio- ja funktiosymboleista (vakiosymboleja on oltava ainakin yksi).

  Huomio. Mikäli polulla P ei esiinny vakiosymboleita, $T\,\forall x\varphi(x)$ ($E\,\exists x\varphi(x)$) tulee hajoittaa käyttäen jotain uutta vakiosymbolia c.

- Muilta osin semanttisen taulun (ja sen polkujen) ristiriitaisuuden ja valmiuden määritelmät säilyvät ennallaan.

Esimerkki. Allaolevan semanttisen taulun kaikki polut ovat valmiit:

1. $T\forall x(P(x) \rightarrow Q(x))$
2. $E(Q(a) \vee Q(b))$
3. $EQ(a)$ $^{2.}$
4. $EQ(b)$ $^{2.}$
5. $T(P(a) \rightarrow Q(a))$ $^{1.\ x/a}$
   - 6. $EP(a)$ $^{5.}$
     - 7. $T(P(b) \rightarrow Q(b))$ $^{1.\ x/b}$
       - 8. $EP(b)$ $^{7.}$
       - 8. $TQ(b)$ $^{7.}$ $\times$
   - 6. $TQ(a)$ $^{5.}$ $\times$

Loogisten ongelmien ratkominen

Taulumenetelmää voidaan käyttää erilaisten loogisten ongelmien ratkomiseen aivan kuten lauselogiikankin tapauksessa.

Määritelmä. Taulu τ on lauseen ϕ *todistus*, jos taulun τ juurisolmuna on $E\phi$ ja τ on ristiriitainen. Jos lauseella ϕ on todistus, on ϕ *teoreema/todistuva* (merkitään $\vdash \phi$).

Väite. $\vdash \phi \iff \models \phi$ (virheettömyys ja täydellisyys).

Määritelmä. Lause ϕ *on johdettavissa* äärellisestä lausejoukosta $\Sigma = \{\phi_1, \ldots, \phi_n\}$ (merkitään $\Sigma \vdash \phi$), joss juurisolmusta

$$E(\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \phi)$$

muodostettu valmis semanttinen taulu on ristiriitainen.

Väite. $\Sigma \vdash \phi \iff \Sigma \models \phi$ (virheettömyys ja täydellisyys).

Esimerkki. Onko $\{P(a), \forall x(P(x) \rightarrow Q(x))\} \vdash Q(a)$?

1. $T\,\forall x(P(x) \rightarrow Q(x))$
2. $T\,P(a)$
3. $E\,Q(a)$
4. $T(P(a) \rightarrow Q(a))$ $^{1.\ x/a}$
   - 5. $E\,P(a)$ $^{4.}$ $\times$
   - 5. $T\,Q(a)$ $^{4.}$ $\times$

Taulu on ristiiriitainen. Lause $Q(a)$ on siis johdettavissa lausejoukosta $\{P(a), \forall x(P(x) \rightarrow Q(x))\}$ ja siten myös lausejoukon looginen seuraus.

Esimerkki. Onko $\{\exists x(P(x) \wedge Q(x))\} \vdash P(a)$?

1. $T\,\exists x(P(x) \wedge Q(x))$
2. $EP(a)$
3. $T(P(b) \wedge Q(b))$ $^{1.\ x/b}$
4. $TP(b)$ $^{3.}$
5. $TQ(b)$ $^{3.}$

Taulu saatiin valmiiksi muttei ristiriitaiseksi. Lause $P(a)$ ei ole johdettavissa lausejoukosta $\{\exists x(P(x) \wedge Q(x))\}$ (eikä myöskään lausejoukon looginen seuraus).

4.3 Ohjeita taulutodistusten laadintaan

- Lauseen rakenne määrää edelleen, mitä taulusääntöä tulee käyttää (jäsennyspuun juuressa oleva konnektiivi).
- Solmujen hajoittamisjärjestyksellä voi usein vaikuttaa taulun kokoon (haarautumista kannattaa välttää).
- Jälkimmäisissä kvanttorisäännöissä esiintyvän termin t tilalle valitaan *hajoittamishetkellä* (eikä myöhemmin) jokin vakio tai funktio- ja vakiosymboleista rakentuva muuttujaton termi.
- Valitsemalla muuttujattomat termit t sopivasti voidaan usein nopeuttaa taulun valmistumista.

Taulutodistusten erityispiirteitä predikaattilogiikan tapauksessa

1. Valitaan muuttujattomaksi termiksi t vakio, joka ei esiinny lauseissa.

Esimerkki. Osoitetaan $\{\forall x P(x)\} \vdash \exists x P(x)$.

1. $T\,\forall x P(x)$
2. $E\,\exists x P(x)$
3. $TP(c)^{1.\ x/c}$
4. $EP(c)^{2.\ x/c}$

$\times$

Huomio. Tämä on perusteltua, koska universumissa U on aina vähintään yksi alkio $a \in U$, joka voidaan nimetä (eli $c^{\mathcal{S}} = a$).

2. Solmu $T\,\forall x \varphi(x)$ (tai $E\,\exists x \varphi(x)$) joudutaan hajoittamaan useasti.

Esimerkki. $\{\forall x P(x)\} \vdash P(a) \wedge P(b)$

1. $T\,\forall x P(x)$
2. $E(P(a) \wedge P(b))$
3. $TP(a)^{1.\ x/a}$
4. $TP(b)^{1.\ x/b}$

5. $EP(a)^{2.}$ $\times$ | 5. $EP(b)^{2.}$ $\times$

1. $T\,\forall x P(x)$
2. $E(P(a) \wedge P(b))$

3. $EP(a)^{2.}$ — 4. $TP(a)^{1.\ x/a}$ $\times$ | 3. $EP(b)^{2.}$ — 4. $TP(b)^{1.\ x/b}$ $\times$

3. Muuttujien korvaaminen sopivilla muuttujattomilla termeillä.

Esimerkki.
$\{\forall x \forall y \forall z (P(x,y) \wedge P(y,z) \rightarrow P(x,z)), P(a,b), P(b,c)\} \vdash P(a,c)$

- Semanttiseen tauluun tulee solmu $T\,\forall x \forall y \forall z (P(x,y) \wedge P(y,z) \rightarrow P(x,z))$, josta voidaan johtaa kvanttorisäännöillä 27 erilaista todeksi merkittyä implikaatiota.
- Ristiriidan johtamisen kannalta olennaisia ovat implikaatioista ne, joissa esiintyy atomisia lauseita $P(a,b)$, $P(b,c)$ ja $P(a,c)$.
- Esimerkin tapauksessa tämä johtaa ensimmäiseksi x:n, y:n ja z:n korvaamiseen vakoilla a, b, ja c (näin saatava implikaatio riittää).
- Muita implikaatioita ei tarvita, ja niiden johtaminen ja mahdollinen hajoittaminen johtaa semanttisen taulun tarpeettomaan kasvuun.

Taulutodistus muodostuu kokonaisuudessaan seuraavaksi:

$$1.\ T\,\forall x \forall y \forall z (P(x,y) \land P(y,z) \rightarrow P(x,z))$$

$$2.\ T P(a,b)$$

$$3.\ T P(b,c)$$

$$4.\ E P(a,c)$$

$$5.\ T\,\forall y \forall z (P(a,y) \land P(y,z) \rightarrow P(a,z))^{1.\ x/a}$$

$$6.\ T\,\forall z (P(a,b) \land P(b,z) \rightarrow P(a,z))^{5.\ y/b}$$

$$7.\ T (P(a,b) \land P(b,c) \rightarrow P(a,c))^{6.\ z/c}$$

$$8.\ E (P(a,b) \land P(b,c))^{7.} \qquad 8.\ T P(a,c)^{7.}\ \times$$

$$9.\ E P(a,b)^{8.}\ \times \qquad 9.\ E P(b,c)^{8.}\ \times$$

4. Solmujen keskinäinen hajoitusjärjestys voi olla ratkaisevassa roolissa.

Esimerkki. $\{\forall x P(x), \forall x (P(x) \rightarrow Q(x)), Q(a)\} \vdash \forall x Q(x)$

$$1.\ T\,\forall x P(x)$$

$$2.\ T\,\forall x (P(x) \rightarrow Q(x))$$

$$3.\ T\,Q(a)$$

$$4.\ E\,\forall x Q(x)$$

$$5.\ E\,Q(c)^{4.\ x/c}$$

$$6.\ T\,P(c)^{1.\ x/c}$$

$$7.\ T (P(c) \rightarrow Q(c))^{2.\ x/c}$$

$$8.\ E\,P(c)^{7.}\ \times \qquad 8.\ T\,Q(c)^{7.}\ \times$$

Kvanttorisekvenssien käsittely

Jatkossa sallimme seuraavien johdettujen taulusääntöjen käytön kvanttorisekvenssien käsittelyssä:

$T\,\forall x_1 \cdots \forall x_n \varphi(x_1,\ldots,x_n)$ $T\varphi(t_1,\ldots,t_n)$	$E\,\exists x_1 \cdots \exists x_n \varphi(x_1,\ldots,x_n)$ $E\varphi(t_1,\ldots,t_n)$
$T\,\exists x_1 \cdots \exists x_n \varphi(x_1,\ldots,x_n)$ $T\varphi(c_1,\ldots,c_n)$	$E\,\forall x_1 \cdots \forall x_n \varphi(x_1,\ldots,x_n)$ $E\varphi(c_1,\ldots,c_n)$

Yllä $c_1,\ldots,c_n$ ovat ao. taulusääntöjen edellyttämiä *uusia vakioita* ja vastaavasti $t_1,\ldots,t_n$ ovat valittuja *muuttujattomia termejä*.

4.4 Systemaattinen taulu

- Lauselogiikan keskeiset päättelyongelmat ovat *ratkeavia*.

Esimerkki. Voidaan konstruoida *deterministinen* Turing-kone T, jonka suoritus pysähtyy syötteeksi annetulla lauselogiikan lauseella φ

1. hyväksyvään tilaan k (kyllä), jos syöte φ on pätevä, ja
2. hylkäävään tilaan e (ei), jos syöte φ ei ole pätevä.

Huomioita.

- Tällainen algoritmi voi perustua esim. totuustaulukkoihin, semanttisiin tauluihin tai resoluutioon.
- Myös looginen ekvivalenssi, looginen seuraavuus ja toteutuvuus ovat lauselogiikan tapauksessa ratkeavia ongelmia.

- Predikaattilogiikka ei ole ratkeava, vaan *puoliratkeava*.

Esimerkki. Lauseen φ pätevyyden tarkastamista varten voidaan konstruoida seuraavanlainen deterministinen Turing-kone T:

1. Jos syöte φ on pätevä, T pysähtyy hyväksyvään tilaan k (kyllä).
2. Jos syöte φ ei ole pätevä, T pysähtyy *joskus* hylkäävään tilaan e (ei) ja *joskus T ei pysähdy lainkaan*.

Huomio. Tällainen algoritmi voi perustua semanttisiin tauluihin:

- Rakentamalla semanttinen taulu tietyllä tavalla *systemaattisesti*, voidaan taata, että taulu saadaan aina ristiriitaiseksi, kun sen juuressa on $E\varphi$ ja φ on pätevä (edellytyksenä täydellisyydelle).

Systemaattisen taulun periaatteita

- Tuotetaan indeksoimalla riittävä määrä uusia vakioita $c_1, c_2, c_3, \ldots$, kun hajoitetaan muotoa $T\,\exists x\psi(x)$ tai $E\,\forall x\psi(x)$ olevia solmuja.
- Tuotetaan tarpeen mukaan muuttujattomia termejä $t_1, t_2, t_3, \ldots$, jotka rakentuvat $E\varphi$:ssä esiintyvistä vakio- ja funktiosymboleista sekä mahdollisesti käyttöönotetuista uusista vakioista $c_1, c_2, c_3, \ldots$.
- Sekvenssin $t_1, t_2, t_3, \ldots$ on oltava *reilu*: jokainen em. symboleista rakentuva muuttujaton termi t esiintyy siinä jonain terminä t_i.
- *Hajoitusten reiluus*: taataan, että taulun keskeneräisillä poluilla esiintyvät hajoittamattomat solmut tulevat hajoitusvuoroon (seuraavan kerran) äärellisen monen muun hajoituksen jälkeen.
- Muotoa $T\,\forall x\psi(x)$ tai $E\,\exists x\psi(x)$ olevia solmuja hajoitetaan järjestyksessä käyttäen muuttujattomia termejä $t_1, t_2, t_3, \ldots$.

Huomio. Muotoa $T\,\forall x\phi(x)$ tai $E\,\exists x\phi(x)$ olevia solmuja sisältäviä polkuja ei välttämättä saada hajoitetuksi äärellisellä askelmäärällä, jolloin valmis taulu muodostuu äärettömäksi (puoliratkeavuuden ilmentymä).

Esimerkki. Kirjoitetaan luonnollisten lukujen $>$-relaatiolle määritelmä. Olkoon $G(x,y) =$ "$x > y$"ja $s(x)$ luvun x seuraaja (eli $x+1$).

- Määritelmä: $\forall x\, G(s(x), x)$ ja $\forall x \forall y (G(x,y) \rightarrow G(s(x), y))$.
- Kysely: onko $G(s(s(s(0))), s(0))$ määritelmän looginen seuraus?

Hajoitetaan systemaattisesti semanttisen taulun solmuja

$$T\,\forall x\, G(s(x), x) \text{ ja } T\,\forall x \forall y (G(x,y) \rightarrow G(s(x), y))$$

käyttäen muuttujattomia termejä: $0, s(0), s(s(0)), s(s(s(0))), \ldots$

- Mikäli solmujen hajoitusjärjestys rikkoo edellä esitettyä reiluusperiaatetta, todistuksen löytyminen ei ole taattu.

Esimerkki. Järjestys, missä hajoitetaan ainoastaan solmua $T\,\forall x\, G(s(x), x)$ em. muuttujattomien termien suhteen, ei ole reilu:

$$1.\ T\,\forall x\, G(s(x), x)$$
$$2.\ T\,\forall x \forall y (G(x,y) \rightarrow G(s(x), y))$$
$$3.\ E\, G(s(s(s(0))), s(0))$$
$$4.\ T\, G(s(0), 0)\ {}^{1.\ x/0}$$
$$5.\ T\, G(s(s(0)), s(0))\ {}^{1.\ x/s(0)}$$
$$6.\ T\, G(s(s(s(0))), s(s(0)))\ {}^{1.\ x/s(s(0))}$$
$$\vdots$$

Esimerkki. Käytetään reilua hajoitusjärjestystä:

1. $T\,\forall x\, G(s(x), x)$
2. $T\,\forall x \forall y (G(x, y) \to G(s(x), y))$
3. $E\, G(s(s(s(0))), s(0))$
4. $T\, G(s(0), 0)$ $^{1.\ x/0}$
5. $T\,\forall y\, (G(0, y) \to G(s(0), y))$ $^{2.\ x/0}$
6. $T(G(0, 0) \to G(s(0), 0))$ $^{5.\ y/0}$

Left branch:

7. $E\, G(0, 0)$
8. $T\, G(s(s(0)), s(0))$ $^{1.\ x/s(0)}$
9. $T\,\forall y (G(s(0), y) \to G(s(s(0)), y))$ $^{2.\ x/s(0)}$

$\vdots$

Right branch:

7. $T\, G(s(0), 0)$
8. $T\, G(s(s(0)), s(0))$ $^{1.\ x/s(0)}$

$\vdots$

Systemaattinen taulu voi tehdä turhaa työtä $\Longrightarrow$ heuristiikka tarpeen!

Esimerkki. Valitsemalla muuttujattomat termit aikaisemmin esitetyillä periaatteilla semanttinen taulu jää huomattavasti pienemmäksi:

1. $T\,\forall x G(s(x), x)$
2. $T\,\forall x \forall y (G(x, y) \to G(s(x), y))$
3. $E\, G(s(s(s(0))), s(0))$
4. $T\,\forall y (G(s(s(0)), y) \to G(s(s(s(0))), y))$ $^{2.\ x/s(s(0))}$
5. $T\,(G(s(s(0)), s(0)) \to G(s(s(s(0))), s(0)))$ $^{5.\ y/s(0)}$

Left branch:

6. $E\, G(s(s(0)), s(0))$
7. $T\, G(s(s(0)), s(0))$ $^{1.\ x/s(0)}$

$\times$

Right branch:

6. $T\, G(s(s(s(0))), s(0))$

$\times$

4.5 Vastamallien konstruointi

- Vastamallin (struktuuri) $\mathcal{S}$ konstruoinnissa voidaan hyödyntää semanttisen taulun ristiriidattomasta polusta saatavia *atomisia lauseita* koskevia totuusarvovaatimuksia $T\,P(t_1, \ldots, t_n)$, $E\,Q(s_1, \ldots, s_m)$, $\ldots$ (t_i:t ja s_j:t ovat muuttujattomia termejä).
- Valitaan riittävän iso universumi U, jotta pystytään antamaan tulkinnat totuusarvovaatimuksissa esiintyville vakio- ja funktiosymboleille.
- Tämän jälkeen valitaan predikaattien tulkinnat totuusarvovaatimusten mukaisesti:
  1. Jos $T\,P(t_1, \ldots, t_n)$ on polulla, $\langle t_1^{\mathcal{S}}, \ldots, t_n^{\mathcal{S}} \rangle \in P^{\mathcal{S}}$.
  2. Jos $E\,Q(s_1, \ldots, s_m)$ on polulla, $\langle s_1^{\mathcal{S}}, \ldots, s_m^{\mathcal{S}} \rangle \notin Q^{\mathcal{S}}$.

- Menettely on käyttökelpoinen erityisesti, jos *valmiin semanttisen taulun* ristiriidaton polku muodostuu *äärelliseksi*:

Esimerkki. Vastamalli $\mathcal{S}$ lauseen $\forall x(P(x) \to Q(x))$ pätevyydelle.

1. $E\,\forall x(P(x) \to Q(x))$
2. $E(P(c) \to Q(c))$ $^{1.\ x/c}$
3. $T(P(c))$ $^{2.}$
4. $E(Q(c))$ $^{2.}$

1. Totuusarvovaatimukset ristiriidattomasta polusta: $T\,P(c)$ ja $E\,Q(c)$.
2. Riittää, että universumiin $U = \{1\}$ otetaan yksi alkio s.e. $c^{\mathcal{S}} = 1$.
3. Totuusarvovaatimusten nojalla: $1 \in P^{\mathcal{S}}$ ja $1 \notin Q^{\mathcal{S}}$.
4. Nämä vaatimukset toteutuvat valinnoilla $P^{\mathcal{S}} = \{1\}$ ja $Q^{\mathcal{S}} = \emptyset$.

Esimerkki. $\{\forall x(P(x) \land Q(x) \to R(x))\} \not\models \forall x(Q(x) \to R(x))$.

1. $T\,\forall x(P(x) \land Q(x) \to R(x))$
2. $E\,\forall x(Q(x) \to R(x))$
3. $E(Q(c) \to R(c))$ $^{2.\ x/c}$
4. $T\,Q(c)$ $^{3.}$
5. $E\,R(c)$ $^{3.}$
6. $T\,(P(c) \land Q(c) \to R(c))$ $^{1.\ x/c}$

7. $E(P(c) \land Q(c))$ 7. $T\,R(c)$ $\times$

8. $E\,P(c)$ 8. $E\,Q(c)$ $\times$

- Tarkastellaan taulun ainoaa ristiriidatonta polkua P.
  - Polulla esiintyy yksi vakiosymboli c muttei funktiosymboleja.
  - Voidaan muodostaa ainoastaan yksi muuttujaton termi eli c itse.
  - Täten solmu $T\,\forall x(P(x) \land Q(x) \to R(x))$ on hajoitettu polulla P, koska polulla P on solmu $T(P(t) \land Q(t) \to R(t))$ jokaista muuttujatonta termiä $t \in \{c\}$ kohti.
  - Polku P on siis valmis.
- Näin ollen taulu on kokonaisuutena myös valmis.
- Polulta P saadaan totuusarvovaatimukset $E\,P(c)$, $T\,Q(c)$ ja $E\,R(c)$.
- Muodostetaan vastamalli $\mathcal{S}$ valitsemalla universumiksi $U = \{1\}$ ja symbolien tulkinnoiksi $c^{\mathcal{S}} = 1$, $P^{\mathcal{S}} = R^{\mathcal{S}} = \emptyset$ ja $Q^{\mathcal{S}} = \{1\}$.

Esimerkki. Joskus äärettömästäkin ristiriidattomasta polusta voi onnistua muodostamaan vastamallin, jolla on *äärellinen* universumi U.

1. $E\,\exists x(P(a) \lor P(f(x)))$
2. $E(P(a) \lor P(f(a)))$ $^{1.\ x/a}$
3. $E\,P(a)$ $^{2.}$
4. $E\,P(f(a))$ $^{2.}$
5. $E(P(a) \lor P(f(f(a))))$ $^{1.\ x/f(a)}$
6. $E\,P(a)$ $^{5.}$
7. $E\,P(f(f(a)))$ $^{5.}$

$\vdots$

Edellytyksenä on muuttujattomien termien tulkinta samalle U:n alkiolle.

- Tarkastellaan taulun ainoata ristiriidatonta polkua P.
- Polku P ei ole valmis, koska solmua $E\,\exists x(P(a) \lor P(f(x)))$ ei ole hajoitettu esim. muuttujattoman termin $t = f(f(a))$ suhteen.
- Polulta P saadaan ääretön sekvenssi totuusarvovaatimuksia
$$E\,P(a),\ E\,P(f(a)),\ E\,P(f(f(a))),\ \ldots$$
- Näiden säännönmukaisuudesta johtuen valitaan universumiksi $U = \{1\}$ ja symbolien tulkinnoiksi $a^{\mathcal{S}} = 1$, $f^{\mathcal{S}} : 1 \mapsto 1$ ja $P^{\mathcal{S}} = \emptyset$.
- Koska taulu ei ollut valmis, lisäksi on syytä todeta totuusmääritelmästä lähtien, että kysymyksessä on todella vastaesimerkki eli $\mathcal{S} \not\models \exists x(P(a) \lor P(f(x)))$.
- Näin muodostettu struktuuri $\mathcal{S}$ muodostaa vastamallin lauseen $\exists x(P(a) \lor P(f(x)))$ pätevyydelle.

5 Tietämyksen esittämisestä

- Tietämyksen esittäminen predikaattilogiikalla
- Ohjeita predikaattien määrittelemiseen
- Nimien yksikäsitteisyys ja kattavuus
- Negatiiviset ehdot ja johtopäätökset

5.1 Tietämyksen esittäminen predikaattilogiikalla

Annettuun järjestelmään liittyvää tietämystä voidaan esittää valitsemalla

- sopiva predikaattilogiikan aakkosto (joukot $\mathcal{P}$, $\mathcal{C}$ ja $\mathcal{F}$) ja
- vastaavaan kieleen $\mathcal{L}$ perustuva lausejoukko $\Sigma \subseteq \mathcal{L}$, jonka lauseet määrittelevät järjestelmän ominaisuudet.

Tarkastellaan määritelmiä vastaavan lausejoukon $\Sigma \subseteq \mathcal{L}$ *loogisten seurausten* joukkoa $\mathrm{Cn}(\Sigma) = \{\phi \in \mathcal{L} \mid \phi \text{ on lause ja } \Sigma \models \phi\}$. Nyt

- Σ muodostaa järjestelmää koskevan *eksplisiittisen tietämyksen* ja
- joukon $\mathrm{Cn}(\Sigma) - \Sigma$ lauseet ovat *implisiittistä tietämystä* eli väittämiä, jotka voidaan päätellä eksplisiittisestä tietämyksestä.

Esimerkki. Kuvataan radioverkon linkkien välityksellä syntyviä yhteyksiä seuraavalla predikaattilogiikan lausejoukolla Σ:

$$
\begin{aligned}
\{\ & \mathsf{linkki}(a,b),\ \mathsf{linkki}(b,c),\ \mathsf{linkki}(d,e),\\
& \forall x\, \mathsf{yhteys}(x,x),\\
& \forall x \forall y (\mathsf{linkki}(x,y) \rightarrow \mathsf{yhteys}(x,y)),\\
& \forall x \forall y (\mathsf{yhteys}(x,y) \rightarrow \mathsf{yhteys}(y,x)),\\
& \forall x \forall y \forall z (\mathsf{yhteys}(x,y) \land \mathsf{yhteys}(y,z) \rightarrow \mathsf{yhteys}(x,z))\ \}.
\end{aligned}
$$

Nyt esim. lause $\mathsf{linkki}(a,b)$ on eksplisiittistä (ylöskirjattua) tietämystä, kun taas esim. lause $\mathsf{yhteys}(c,a)$ lukeutuu lausejoukon Σ loogisena seurauksena osaksi implisiittistä tietämystä.

Tietämyksen esittämisen problematiikkaa (kertaus)

- Kaikki struktuurit ovat tyhjän lausejoukon $\emptyset$ malleja, joten $\mathrm{Cn}(\emptyset)$ on itse asiassa pätevien lauseiden joukko.
- Enemmän lauseita $\Rightarrow$ vähemmän malleja $\Rightarrow$ enemmän loogisia seurauksia: siis $\Sigma_1 \subseteq \Sigma_2 \Rightarrow \mathrm{Cn}(\Sigma_1) \subseteq \mathrm{Cn}(\Sigma_2)$ (*monotonisuus*).
- Jos lausejoukko tulee ristiriitaiseksi, sillä ei ole yhtään mallia ja kaikki lauseet ovat tällöin lausejoukon loogisia seurauksia.

$\Longrightarrow$ Tavoitteena rajata predikaattilogiikan lausejoukolla $\Sigma \subseteq \mathcal{L}$ mahdollisten asiaintilojen (eli lausejoukon Σ mallien) joukkoa sopivasti siten, että saadaan halutut loogiset seuraukset.

Esimerkki. Palataan radioverkkoesimerkin lausejoukkoon Σ, jonka osalta voidaan todeta esim. $\Sigma \not\models$ yhteys(a, e).

- Kirjataan tälle vastamalliksi esim. seuraava struktuuri $\mathcal{S}$:

  Universumi $U = \{1, 2, 3, 4, 5\}$.

  Vakioiden tulkinnat: $a^{\mathcal{S}} = 1$, $b^{\mathcal{S}} = 2$, $c^{\mathcal{S}} = 3$, $d^{\mathcal{S}} = 4$ ja $e^{\mathcal{S}} = 5$.

  Predikaattien tulkinnat:

  — linkki$^{\mathcal{S}} = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 4,5\rangle\}$ ja

  — yhteys$^{\mathcal{S}} = \left\{ \begin{array}{l} \langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,1\rangle, \langle 2,2\rangle, \langle 2,3\rangle, \\ \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle, \langle 4,4\rangle, \langle 4,5\rangle, \langle 5,4\rangle, \langle 5,5\rangle \end{array} \right\}$.

- Esim. lisäämällä lause linkki(d, c) saadaan laajennettu lausejoukko $\Sigma' = \Sigma \cup \{$linkki$(d, c)\}$, jolle $\Sigma' \models$ yhteys(a, e).

- Huomaa, että $\mathcal{S} \not\models$ linkki(d, c), joten vastamallimme rajautuu pois.

5.2 Ohjeita predikaattien määrittelemiseen

- Tavoitteena kirjoittaa annetulle predikaatille P (ja siten myös sen tulkintana olevalle relaatiolle) määritelmä joidenkin muiden predikaattien avulla.

- Mielivaltainen predikaattilogiikan kaava ϕ voidaan saattaa muotoon

$$\mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \cdots \mathcal{Q}_n x_n \psi$$

  missä kukin kvanttori $\mathcal{Q}_i$ on joko $\forall$ tai $\exists$, ja kaava ψ on konjunktiivisessa normaalimuodossa eikä sisällä kvanttoreita.

- Yllä $\psi = \psi_1 \wedge \cdots \wedge \psi_m$, missä kukin ψ_i on literaalien disjunktio

$$\begin{aligned} & \neg Q_1(\vec{t_1}) \vee \cdots \vee \neg Q_k(\vec{t_k}) \vee P_1(\vec{s_1}) \vee \cdots \vee P_l(\vec{s_l}) \\ \equiv \quad & Q_1(\vec{t_1}) \wedge \cdots \wedge Q_k(\vec{t_k}) \rightarrow P_1(\vec{s_1}) \vee \cdots \vee P_l(\vec{s_l}). \end{aligned}$$

- Predikaattilogiikalla annetut määritelmät ovat korkeintaan yhtä monimutkaisia kuin edellä kuvattu normaalimuoto.

- Jos jokainen kvanttori $\mathcal{Q}_i$ on universaalikvanttori $\forall$, $m = 1$ ja $l = 1$, saamme erikoistapauksena muotoa

$$\forall x_1 \forall x_2 \cdots \forall x_n (Q_1(\vec{t_1}) \wedge \cdots \wedge Q_k(\vec{t_k}) \rightarrow P(\vec{t}))$$

  olevia lauseita, missä predikaattien $Q_1, \ldots, Q_k$ ja P argumentteina on termijonot $\vec{t_1}, \ldots, \vec{t_k}$ ja $\vec{t}$, jotka rakentuvat siis vakiosymboleista, muuttujasymboleista $x_1, \ldots, x_n$ ja funktiosymboleista.

- Määritelmiä voidaan usein kirjoittaa tähän muotoon: mietitään millä ehdoilla $Q_1(\vec{t_1}), \ldots, Q_k(\vec{t_k})$ voidaan päätellä predikaattia P koskeva väittämä $P(\vec{t})$. Ko. muotoa olevia lauseita saatetaan tarvita useita.

Esimerkki. Olkoon annettuna predikaatit

1. sairastaa(x) = "henkilö x on sairas" ja
2. tapaa(x, y) = "henkilö x tapaa henkilön y".

Tarkoituksena on määritellä näiden avulla predikaatti

tartuntavaarassa(x) = "henkilö x on tartuntavaarassa".

Kysymys: millä ehdoilla jonkin henkilö on tartuntavaarassa?

1. Jos henkilö tapaa jonkun sairaan henkilön.
2. Jos henkilö tapaa jonkun toisen tartuntavaarassa olevan henkilön.

Yritetään kirjoittaa nämä edellä esitetyn mukaisesti muotoon

$$\forall x \forall y \ldots (Q_1(\vec{t_1}) \wedge \ldots \wedge Q_k(\vec{t_k}) \rightarrow \text{tartuntavaarassa}(x)).$$

- Näin saadaan muodostettua lauseet

  $\forall x \forall y(\mathsf{tapaa}(x,y) \land \mathsf{sairastaa}(y) \rightarrow \mathsf{tartuntavaarassa}(x))$ ja

  $\forall x \forall y(\mathsf{tapaa}(x,y) \land \mathsf{tartuntavaarassa}(y) \rightarrow \mathsf{tartuntavaarassa}(x))$.

- Kysymyksessä on itse asiassa tartuntavaarassa-predikaatin induktiivinen (rekursiivinen) määritelmä. Lauseista ensimmäinen vastaa perustapausta ja jälkimmäinen induktioaskelta.
- Lisäksi voidaan todeta tapaamiset symmetrisiksi:

$$\forall x \forall y(\mathsf{tapaa}(x,y) \rightarrow \mathsf{tapaa}(y,x)).$$

- Universumin voidaan ajatella koostuvan pelkästään henkilöistä (eli edellä annetut lauseet puhuvat kaikista henkilöistä).

Määritelmien käyttö konkreettisessa päättelyssä

Esimerkki. Lisätään edellä johdettuun tartuntavaarassa-predikaatin määritelmään tietokanta, jossa kuvataan tapaamiset ja sairastamiset:

Näin saadaan lausejoukko

$$\begin{aligned}\Sigma = \{\ & \forall x \forall y(\mathsf{tapaa}(x,y) \land \mathsf{sairastaa}(y) \rightarrow \mathsf{tartuntavaarassa}(x)),\\ & \forall x \forall y(\mathsf{tapaa}(x,y) \land \mathsf{tartuntavaarassa}(y) \rightarrow \mathsf{tartuntavaarassa}(x)),\\ & \forall x \forall y(\mathsf{tapaa}(x,y) \rightarrow \mathsf{tapaa}(y,x)),\\ & \mathsf{tapaa}(\mathsf{Lyyli},\mathsf{Hemmo}),\ \mathsf{tapaa}(\mathsf{Lyyli},\mathsf{Erkki}),\ \mathsf{sairastaa}(\mathsf{Erkki})\ \}.\end{aligned}$$

- Kyseisessä asetelmassa saadaan $\Sigma \models \mathsf{tartuntavaarassa}(\mathsf{Lyyli}) \land \mathsf{tartuntavaarassa}(\mathsf{Hemmo})$.
- Kokeile tämän osoittamista semanttisella taululla!

Määritelmien tarkkuudesta

- Olkoon H tarkasteltavan predikaattilogiikan kielen $\mathcal{L}$ muuttujattomien termien joukko.
- Tavoitteenamme on siis kirjoittaa predikaatin $P \in \mathcal{P}_n$ määrittelevä lausejoukko Σ_P, kun lähtökohtana on tieto predikaatin P tarkoittamasta relaatiosta $P^* \subseteq H^n$.
- Määritelmää Σ_P voidaan pitää riittävän tarkkana, jos kaikille muuttujattomille termeille $t_1 \in H, \ldots, t_n \in H$ pätee seuraavaa: $\langle t_1, \ldots, t_n \rangle \in P^* \iff \Sigma_P \models P(t_1, \ldots, t_n)$.
- Tälläinen *positivistinen* määritelmä ei ole välttämättä *täydellinen* eli määritelmä ei takaa, että $\Sigma_P \models \neg P(t_1, \ldots, t_n)$ mikäli $\langle t_1, \ldots, t_n \rangle \notin P^*$.

Esimerkki. Palataan taas radioverkkoesimerkin lausejoukkoon $\Sigma =$

$$\begin{aligned}\{\ & \mathsf{linkki}(a,b),\ \mathsf{linkki}(b,c),\ \mathsf{linkki}(d,e),\\ & \forall x\ \mathsf{yhteys}(x,x),\\ & \forall x \forall y(\mathsf{linkki}(x,y) \rightarrow \mathsf{yhteys}(x,y)),\\ & \forall x \forall y(\mathsf{yhteys}(x,y) \rightarrow \mathsf{yhteys}(y,x)),\\ & \forall x \forall y \forall z(\mathsf{yhteys}(x,y) \land \mathsf{yhteys}(y,z) \rightarrow \mathsf{yhteys}(x,z))\ \}.\end{aligned}$$

- Määritelmän lähtökohtana on relaatio $\mathsf{linkki}^* = \{\langle a,b\rangle, \langle b,c\rangle, \langle d,e\rangle\}$.
- Yhteys-predikaatin määritelmää voidaan pitää riittävän tarkkana, koska tavoiteltu relaatio $\mathsf{yhteys}^* = \{\langle a,a\rangle, \langle a,b\rangle, \langle a,c\rangle, \langle b,a\rangle, \langle b,b\rangle, \langle b,c\rangle, \langle c,a\rangle, \langle c,b\rangle, \langle c,c\rangle, \langle d,d\rangle, \langle d,e\rangle, \langle e,d\rangle, \langle e,e\rangle\}$ saadaan määritelmän Σ loogisina seurauksina edellä kuvatulla tavalla.
- Määritelmä ei ole täydellinen, koska esim. $\Sigma \not\models \neg\mathsf{yhteys}(a,e)$.

Tyypitetyt kvanttorit

- Usein on mielekästä ajatella universumin koostuvan tyypiltään erilaisista alkioista.
- Tällöin syntyy tarve rajata kvantifiointia koskemaan ainoastaan tiettyä tyyppiä T olevia alkioita seuraavaan tapaan:
$$\forall x \in T : \phi(x) \text{ ja } \exists x \in T : \phi(x).$$
- Tyyppi T voidaan esittää yksipaikkaisen predikaatin avulla:
$$T(x) = \text{“alkio } x \text{ on tyyppiä } T\text{”}.$$
- Tyypitetyt kvanttorit ilmaistaan predikaattilogiikassa seuraavasti:
$$\forall x(T(x) \rightarrow \phi(x)) \text{ ja } \exists x(T(x) \wedge \phi(x)).$$

Esimerkki. Lisätään tartuntavaaraesimerkkiin tyyppipredikaatteja:

- Määritellään predikaatit henkilöiden ja tautien erottelemiseksi: henkilö(x) = “x on henkilö” ja tauti(x) = “x on tauti”.
- Määritellään predikaatit ilman tyyppi-informaatiota:
  - tapaa(x, y) = “x tapaa y:n”,
  - sairastaa(x, y) = “x sairastaa y:tä” ja
  - tartuntavaarassa(x, y) = “x on vaarassa sairastua y:hyn”.
- Lauseet saadaan nyt seuraavaan muotoon:

$$\begin{aligned}
&\forall x \forall y \forall z(\text{henkilö}(x) \wedge \text{henkilö}(y) \wedge \text{tapaa}(x, y) \wedge \\
&\qquad \text{tauti}(z) \wedge \text{sairastaa}(y, z) \rightarrow \text{tartuntavaarassa}(x, z)) \text{ ja} \\
&\forall x \forall y \forall z(\text{henkilö}(x) \wedge \text{henkilö}(y) \wedge \text{tapaa}(x, y) \wedge \text{tauti}(z) \wedge \\
&\qquad \text{tartuntavaarassa}(y, z) \rightarrow \text{tartuntavaarassa}(x, z)).
\end{aligned}$$

- Tyypeillä voi olla erilaisia suhteita:
  - Erillisyys: $\forall x \neg(\text{henkilö}(x) \wedge \text{tauti}(x))$.
  - Kattavuus: $\forall x(\text{henkilö}(x) \vee \text{tauti}(x))$.
  - Alityyppi: $\forall x(\text{rokko}(x) \rightarrow \text{tauti}(x))$.
- Yksi mahdollisuus on tyypittää predikaatit erikseen:

$$\forall x \forall y(\text{tapaa}(x, y) \rightarrow \text{henkilö}(x) \wedge \text{henkilö}(y))$$
$$\forall x \forall y(\text{sairastaa}(x, y) \rightarrow \text{henkilö}(x) \wedge \text{tauti}(y))$$

- Tällöin varsinainen tartuntavaarassa-predikaatin määritelmä voidaan kirjoittaa yksinkertaisemmin ilman tyyppi-informaatiota:

$$\forall x \forall y \forall z(\text{tapaa}(x, y) \wedge \text{sairastaa}(y, z) \rightarrow \text{tartuntavaarassa}(x, z)) \text{ ja}$$
$$\forall x \forall y \forall z(\text{tapaa}(x, y) \wedge \text{tartuntavaarassa}(y, z) \rightarrow \text{tartuntavaarassa}(x, z)).$$

Muodoltaan monimutkaisempia määritelmiä

- Edellä otettiin lähtökohdaksi muotoa
$$\forall x_1 \forall x_2 \cdots \forall x_n(Q_1(\vec{t_1}) \wedge \cdots \wedge Q_k(\vec{t_k}) \rightarrow P(\vec{t}))$$
olevat määritelmät. Näiden ilmaisuvoima ei ole aina riittävä.
- Joissain tilanteissa tarvitaan eksistentiaalista kvantifiointia:
$$\begin{aligned}
& \forall x(\text{solmu}(x) \rightarrow \exists y(\text{väri}(y) \wedge \text{väritetty}(x, y))) \\
\equiv \quad & \forall x \exists y(\text{solmu}(x) \rightarrow \text{väri}(y) \wedge \text{väritetty}(x, y)).
\end{aligned}$$
- Implikaation seurauksena voi olla myös atomien disjunktio $P_1(\vec{s_1}) \vee \cdots \vee P_l(\vec{s_l})$ pelkän atomin $P(\vec{t})$ sijaan:
$$\forall x(\text{kaksiarvoinen}(x) \rightarrow \text{nolla}(x) \vee \text{yksi}(x)).$$

Huomio. Edellä oli keskeistä vaihtoehtoisuuden ilmaiseminen.

5.3 Nimien yksikäsitteisyys ja kattavuus

- Rajoitutaan jatkossa predikaattilogiikan kieliin $\mathcal{L}$, joissa ei ole funktiosymboleita ja ainoastaan äärellinen määrä vakiosymboleita.
- Predikaattilogiikassa struktuurin $\mathcal{S}$ määritelmä ja tapa jolla vakiosymbolit tulkitaan $\mathcal{S}$:ssa mahdollistavat, että
  1. jokin universumin U alkio $a \in U$ on useamman vakion $c_1, \ldots, c_n$ $(n > 1)$ nimeämä: $c_1^{\mathcal{S}} = \ldots = c_n^{\mathcal{S}} = a$.
  2. jokin universumin U alkio $a \in U$ ei ole minkään vakion nimeämä (eli kaikille vakiosymboleille c pätee $c^{\mathcal{S}} \neq a$).
- Tietämyksen esittämisen kannalta tällainen mahdollisuus muodostuu usein jopa turhaksi vapausasteeksi.
- Nimeäminen voidaan pakottaa yksikäsitteiseksi lauseita lisäämällä.

Nimien yksikäsitteisyys

- Vastaava käsite englanniksi on *unique names assumption* (UNA).
- Kun kielessä on äärellinen määrä vakiosymboleita $c_1, \ldots, c_n$, riittää lisätä muotoa
$$\neg(c_i = c_j)$$
olevat lauseet, missä $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n\}$ ja $i < j$.
- Lauseita tarvitaan neliöllinen määrä (yhteensä $\frac{n^2-n}{2}$ kappaletta).

Esimerkki. Olkoon kielessä $\mathcal{L}$ vakiosymbolit Lyyli, Hemmo ja Erkki. Yksikäsitteisten nimien oletus ilmaistaan seuraavasti:

$$\neg(\text{Lyyli} = \text{Hemmo}),\ \neg(\text{Lyyli} = \text{Erkki}) \text{ ja } \neg(\text{Hemmo} = \text{Erkki}).$$

Esimerkki. Tarkastellaan lausejoukon

$$\Sigma_{\text{UNA}} = \{\neg(\text{Lyyli} = \text{Hemmo}), \neg(\text{Lyyli} = \text{Erkki}), \neg(\text{Hemmo} = \text{Erkki})\}$$

malleja $\mathcal{S}_i$, kun universumina U_i on joukko henkilöitä $h_1, h_2, \ldots$.

U_i	Lyyli$^{\mathcal{S}_i}$	Hemmo$^{\mathcal{S}_i}$	Erkki$^{\mathcal{S}_i}$
$\{h_1, h_2, h_3\}$	h_1	h_2	h_3
$\{h_1, h_2, h_3\}$	h_1	h_3	h_2
$\{h_1, h_2, h_3\}$	h_2	h_1	h_3
$\{h_1, h_2, h_3\}$	h_2	h_3	h_1
$\{h_1, h_2, h_3\}$	h_3	h_1	h_2
$\{h_1, h_2, h_3\}$	h_3	h_2	h_1
$\{h_1, h_2, h_3, h_4\}$	h_1	h_2	h_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\Longrightarrow$ Universumissa oltava vähintään 3 henkilöä.

Nimien kattavuus

- Vastaava käsite englanniksi on *domain closure assumption* (DCA).
- Kun kielessä on äärellinen määrä vakiosymboleita $c_1, \ldots, c_n$, riittää lisätä seuraavaa muotoa oleva lause:
$$\forall x(x = c_1 \vee \cdots \vee x = c_n).$$
- Tarvittavan lauseen pituus riippuu lineaarisesti vakioiden lukumäärästä n.

Esimerkki. Edellisen esimerkin mukaisessa kielessä tarvitaan lause

$$\forall x(x = \text{Lyyli} \vee x = \text{Hemmo} \vee x = \text{Erkki}).$$

Esimerkki. Tarkastellaan lausejoukon

$$\Sigma_{\mathrm{DCA}} = \{\forall x(x = \text{Lyyli} \lor x = \text{Hemmo} \lor x = \text{Erkki})\}$$

malleja $\mathcal{S}_i$, kun universumina U_i on joukko henkilöitä $h_1, h_2, \ldots$.

U_i	$\text{Lyyli}^{\mathcal{S}_i}$	$\text{Hemmo}^{\mathcal{S}_i}$	$\text{Erkki}^{\mathcal{S}_i}$
$\{h_1\}$	h_1	h_1	h_1
$\{h_1, h_2\}$	h_1	h_1	h_2
$\{h_1, h_2\}$	h_1	h_2	h_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{h_1, h_2, h_3\}$	h_1	h_2	h_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\{h_1, h_2, h_3\}$	h_3	h_2	h_1

$\Longrightarrow$ Universumissa voi olla korkeintaan 3 henkilöä.

Esimerkki. Tarkastellaan vielä edeltävien lausejoukkojen unionin

$$\begin{aligned}\Sigma_{\mathrm{UNA}} \cup \Sigma_{\mathrm{DCA}} = \{\ & \neg(\text{Lyyli} = \text{Hemmo}), \neg(\text{Lyyli} = \text{Erkki}),\\ & \neg(\text{Hemmo} = \text{Erkki}),\\ & \forall x(x = \text{Lyyli} \lor x = \text{Hemmo} \lor x = \text{Erkki})\ \}\end{aligned}$$

malleja $\mathcal{S}_i$, kun universumina U_i on joukko henkilöitä $h_1, h_2, \ldots$.

U_i	$\text{Lyyli}^{\mathcal{S}_i}$	$\text{Hemmo}^{\mathcal{S}_i}$	$\text{Erkki}^{\mathcal{S}_i}$
$\{h_1, h_2, h_3\}$	h_1	h_2	h_3
$\{h_1, h_2, h_3\}$	h_1	h_3	h_2
$\{h_1, h_2, h_3\}$	h_2	h_1	h_3
$\{h_1, h_2, h_3\}$	h_2	h_3	h_1
$\{h_1, h_2, h_3\}$	h_3	h_1	h_2
$\{h_1, h_2, h_3\}$	h_3	h_2	h_1

$\Longrightarrow$ Universumissa on oltava täsmälleen 3 henkilöä.

Yhtäsuuruuspredikaatin määritelmä

Jos yhtäsuuruuspredikaattia sisältäviä lauseita käytetään määritelmissä, seuraavat aksiomat saattavat olla tarpeen esim. todistuksissa.

1. Refleksiivisyys: $\forall x(x = x)$.
2. Symmetrisyys: $\forall x \forall y((x = y) \rightarrow (y = x))$.
3. Transitiivisuus: $\forall x \forall y \forall z((x = y) \land (y = z) \rightarrow (x = z))$.
4. Sijoitettavuus (kaikille predikaateille $P \in \mathcal{P}_n$):

$$\begin{aligned}&\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n\\ &\quad (P(x_1, \ldots, x_n) \land (x_1 = y_1) \land \ldots \land (x_n = y_n) \rightarrow P(y_1, \ldots, y_n)).\end{aligned}$$

Osoita $\{P(a), (b = a)\} \models P(b)$ näiden ja semanttisen taulun avulla!

5.4 Negatiiviset ehdot ja johtopäätökset

- Tarkastellaan muotoa

$$\forall x_1 \forall x_2 \cdots \forall x_n (Q_1(\vec{t_1}) \land \cdots \land Q_k(\vec{t_k}) \rightarrow P(\vec{t}))$$

  olevien määritelmien yleistämistä tapaukseen, missä sallitaan atomien $Q_i(\vec{t_i})$ lisäksi myös negatiivisia literaaleja $\neg Q_i(\vec{t_i})$.

- Negatiivinen ehto $\neg Q_i(\vec{t_i})$ voidaan muuntaa positiiviseksi vaihtoehdoksi $Q_i(\vec{t_i})$ seuraukselle $P(\vec{t})$.

  Esimerkki.

$$\begin{aligned}&\forall x(\neg\text{sairastaa}(x) \land \neg\text{tartuntavaarassa}(x) \rightarrow \text{turvassa}(x))\\ \equiv\ &\forall x(\text{sairastaa}(x) \lor \text{tartuntavaarassa}(x) \lor \text{turvassa}(x)).\end{aligned}$$

- Jotta negatiiviset ehdot tulisivat määritellyiksi, määritelmistä tulisi seurata loogisesti $\neg Q_i(\vec{t_i})$ mikäli $Q_i(\vec{t_i})$ ei ole looginen seuraus.

Määritelmien täydellisyys

Määritelmä. Predikaatin $P \in \mathcal{P}_n$ määritelmä $\Sigma_P \subseteq \mathcal{L}$ on *täydellinen*, jos

$$\Sigma_P \models P(t_1, \ldots, t_n) \text{ tai } \Sigma_P \models \neg P(t_1, \ldots, t_n).$$

kaikille kielen $\mathcal{L}$ muuttujattomille termeille $t_1, \ldots, t_n$.

Huomioita.

- Jos predikaatin $P \in \mathcal{P}_n$ määritelmä Σ_P on ristiriitainen (eli sillä ei ole malleja), se on triviaalisti täydellinen: tällöin sekä $\Sigma_P \models P(t_1, \ldots, t_n)$ että $\Sigma_P \models \neg P(t_1, \ldots, t_n)$ kaikille muuttujattomille termeille $t_1 \in H, \ldots, t_n \in H$.
- Jos predikaatin $P \in \mathcal{P}_n$ määritelmä Σ_P on täydellinen ja $\Sigma_P \not\models P(t_1, \ldots, t_n)$ joillekin muuttujattomille termeille $t_1 \in H$, $\ldots, t_n \in H$, niin Σ_P ei ole ristiriitainen ja $\Sigma_P \models \neg P(t_1, \ldots, t_n)$.

Esimerkki. Tarkastellaan *muunnelmaa* tartuntavaara-esimerkistä:

$$\begin{aligned}\Sigma = \{\ & \forall x \forall y(\text{tapaa}(x,y) \land \text{sairastaa}(y) \rightarrow \text{tartuntavaarassa}(x)),\\ & \forall x(\neg\text{sairastaa}(x) \land \neg\text{tartuntavaarassa}(x) \rightarrow \text{turvassa}(x)),\\ & \text{tapaa}(\text{Lyyli}, \text{Hemmo}),\ \text{tapaa}(\text{Hemmo}, \text{Lyyli}),\\ & \text{tapaa}(\text{Lyyli}, \text{Erkki}),\ \text{tapaa}(\text{Erkki}, \text{Lyyli}),\ \text{sairastaa}(\text{Erkki})\ \}.\end{aligned}$$

- Nyt $\Sigma \models \text{tartuntavaarassa}(\text{Lyyli})$, $\Sigma \not\models \text{tartuntavaarassa}(\text{Hemmo})$ ja $\Sigma \not\models \neg\text{tartuntavaarassa}(\text{Hemmo})$.
- Täten Σ ei ole täydellinen määritelmä tartuntavaarassa-predikaatille.
- Jotta näin olisi, määritelmästä tulisi seurata loogisesti $\neg\text{tartuntavaarassa}(\text{Hemmo})$ ja $\neg\text{tartuntavaarassa}(\text{Erkki})$.
- Kyseinen määritelmä Σ ei ole myöskään täydellinen muille ao. kielen predikaateille (tapaa, sairastaa, turvassa ja $=$).

Predikaatin määritelmän täydentäminen

Esimerkki. Täydennetään edellisen esimerkin predikaattien määritelmät.

1. Yhtäsuuruuspredikaatin osalta riittää todeta nimien yksikäsitteisyys:

$$\neg(\text{Lyyli} = \text{Hemmo}),\ \neg(\text{Lyyli} = \text{Erkki}),\ \neg(\text{Hemmo} = \text{Erkki}) \text{ ja}$$
$$\forall x(x = \text{Lyyli} \lor x = \text{Hemmo} \lor x = \text{Erkki}).$$

2. Predikaateille tapaa ja sairastaa voidaan kirjoittaa tiiviit esitykset yhtäsuuruuspredikaatin avulla: $\forall x(\text{sairastaa}(x) \leftrightarrow x = \text{Erkki})$ ja

$$\begin{aligned}\forall x(\text{tapaa}(x,y) \leftrightarrow\ & (x = \text{Lyyli} \land y = \text{Hemmo}) \lor\\ & (x = \text{Hemmo} \land y = \text{Lyyli}) \lor\\ & (x = \text{Lyyli} \land y = \text{Erkki}) \lor\\ & (x = \text{Erkki} \land y = \text{Lyyli})).\end{aligned}$$

3. Predikaattien tartuntavaarassa ja turvassa määritelmät voidaan kirjoittaa vastaavasti ekvivalensseiksi:

$$\forall x(\text{tartuntavaarassa}(x) \leftrightarrow \exists y(\text{tapaa}(x,y) \land \text{sairastaa}(y))) \text{ ja}$$
$$\forall x(\text{turvassa}(x) \leftrightarrow \neg\text{sairastaa}(x) \land \neg\text{tartuntavaarassa}(x)).$$

- Täydennetyillä määritelmillä on haluamamme loogiset seuraukset:

sairastaa	tartuntavaarassa	turvassa
¬sairastaa(Lyyli)	tartuntavaarassa(Lyyli)	¬turvassa(Lyyli)
¬sairastaa(Hemmo)	¬tartuntavaarassa(Hemmo)	turvassa(Hemmo)
sairastaa(Erkki)	¬tartuntavaarassa(Erkki)	¬turvassa(Erkki)

- Predikaattien täydentäminen ei valitettavasti tuota haluttua lopputulosta *rekursiivisten* määritelmien tapauksessa, kuten seuraavassa esimerkissä osoitetaan.

Esimerkki. Tarkastellaan vastaavaa konstruktiota lausejoukolle

$$
\begin{aligned}
\Sigma = \{ \ & \mathsf{tuntee1}(\mathsf{Lyyli}, \mathsf{Lyyli}), \mathsf{tuntee1}(\mathsf{Hemmo}, \mathsf{Hemmo}), \\
& \forall x \forall y(\mathsf{tuntee1}(x, y) \rightarrow \mathsf{tuntee2}(x, y)), \\
& \forall x \forall y(\mathsf{tuntee2}(y, x) \rightarrow \mathsf{tuntee2}(x, y)) \ \}.
\end{aligned}
$$

- Rekursiivisesti määritellyn predikaatin tuntee2 tarkoituksena on täydentää predikaatti tuntee1 *symmetriseksi*.
- Täydennettynä määritelmät saadaan muotoon

$$
\begin{aligned}
\Sigma' = \{ \ & \neg(\mathsf{Lyyli} = \mathsf{Hemmo}), \quad \forall x(x = \mathsf{Lyyli} \lor x = \mathsf{Hemmo}), \\
& \forall x \forall y(\mathsf{tuntee1}(x, y) \leftrightarrow (x = \mathsf{Lyyli} \land y = \mathsf{Lyyli}) \lor \\
& \qquad (x = \mathsf{Hemmo} \land y = \mathsf{Hemmo})), \\
& \forall x \forall y(\mathsf{tuntee2}(x, y) \leftrightarrow \mathsf{tuntee1}(x, y) \lor \mathsf{tuntee2}(y, x)) \ \}.
\end{aligned}
$$

- Yllättäen täydennetyistä määritelmistä ei seuraa loogisesti $\neg\mathsf{tuntee2}(\mathsf{Lyyli}, \mathsf{Hemmo})$ eikä $\neg\mathsf{tuntee2}(\mathsf{Hemmo}, \mathsf{Lyyli})$.
- Lausejoukolla Σ' on seuraava epäintuitiivinen malli $\mathcal{S}$:
  Universumi $U = \{h_1, h_2\}$,
  $\mathsf{Lyyli}^{\mathcal{S}} = h_1$, $\mathsf{Hemmo}^{\mathcal{S}} = h_2$,
  $\mathsf{tuntee1}^{\mathcal{S}} = \{\langle h_1, h_1\rangle, \langle h_2, h_2\rangle\}$ ja
  $\mathsf{tuntee2}^{\mathcal{S}} = \{\langle h_1, h_1\rangle, \langle h_1, h_2\rangle, \langle h_2, h_1\rangle, \langle h_2, h_2\rangle\}$.
- Kyseinen struktuuri $\mathcal{S}$ on vastamalli, koska

$$\mathcal{S} \not\models \neg\mathsf{tuntee2}(\mathsf{Lyyli}, \mathsf{Hemmo}) \text{ ja } \mathcal{S} \not\models \neg\mathsf{tuntee2}(\mathsf{Hemmo}, \mathsf{Lyyli}).$$

Huomio. Tentissä ei edellytetä täydellisten määritelmien kirjoittamista predikaateille (ellei tätä sitten erikseen jossain yksinkertaisessa tapauksessa pyydetä).

6 Herbrandin teoreema

- Herbrand-universumit
- Herbrand-struktuurit ja -mallit
- Herbrandin teoreema
- Lause- ja predikaattilogiikan suhteesta

6.1 Herbrand-universumit

Määritelmä. Predikaattilogiikan kielen $\mathcal{L}$ *Herbrand-universumi* H on niiden muuttujattomien termien t joukko, jotka ovat muodostettavissa kielen $\mathcal{L}$ vakio- ja funktiosymboleista.

Esimerkki. Olkoon kielessä $\mathcal{L}$ ainoastaan yksi vakiosymboli c ja yksi funktiosymboli $f \in \mathcal{F}_2$.

Herbrand-universumiksi saadaan muuttujattomien termien joukko

$$H = \{c, f(c, c), f(f(c, c), c), f(c, f(c, c)), f(f(c, c), f(c, c)), \ldots\}.$$

Huomio. Jos kielessä $\mathcal{L}$ ei ole funktiosymboleita ja ainoastaan äärellinen määrä vakioita, Herbrand-universumi H jää tällöin äärelliseksi.

- Herbrand-universumi voidaan määritellä myös annetusta lausejoukosta Σ lähtien.
- Jos lausejoukossa Σ ei esiinny yhtään vakiosymbolia, Herbrand-universumiin valitaan ainakin yksi vakiosymboli c (struktuurien määritelmän mukaan universumit ovat aina ei-tyhjiä).

Määritelmä. Lausejoukon Σ Herbrand-universumi H on niiden muuttujattomien termien t joukko, jotka ovat muodostettavissa lausejoukossa Σ esiintyvistä vakio- ja funktiosymboleista.

Esimerkki. Lausejoukon $\Sigma = \{\forall x P(x, f(x))\}$ Herbrand-universumi on

$$H = \{c, f(c), f(f(c)), \ldots\} = \{f^n(c) \mid n \geq 0\}.$$

6.2 Herbrand-struktuurit ja -mallit

Määritelmä. Kielen $\mathcal{L}$ *Herbrand-struktuuri* on struktuuri $\mathcal{H}$, jonka

1. universumina on kielen $\mathcal{L}$ Herbrand-universumi H,
2. jokaisen vakiosymbolin $c \in \mathcal{C}$ tulkintana $c^{\mathcal{H}}$ on c itse,
3. jokaisen muuttujasymbolin $x \in \mathcal{V}$ tulkintana $x^{\mathcal{H}}$ on jokin muuttujaton termi $t \in H$,
4. jokaisen funktiosymbolin $f \in \mathcal{F}_n$ tulkintana on funktio $f^{\mathcal{H}} : H^n \to H$, joka kuvaa muuttujattomat termit $t_1 \in H, \ldots, t_n \in H$ muuttujattomaksi termiksi $f(t_1, \ldots, t_n) \in H$, ja
5. jokaisen predikaattisymbolin $P \in \mathcal{P}_n$ tulkintana on $P^{\mathcal{H}} \subseteq H^n$.

Lauseen $\phi \in \mathcal{L}$ totuusarvo Herbrand-struktuurissa $\mathcal{H}$ lasketaan predikaattilogiikan totuusmääritelmän mukaisesti.

Määritelmä. Kielen $\mathcal{L}$ Herbrand-struktuuri $\mathcal{H}$ on

1. lauseen $\phi \in \mathcal{L}$ *Herbrand-malli* $\iff$ $\mathcal{H} \models \phi$, ja
2. lausejoukon $\Sigma \subseteq \mathcal{L}$ *Herbrand-malli* $\iff$ $\mathcal{H} \models \sigma$ kaikille $\sigma \in \Sigma$.

Esimerkki. Tarkastellaan lausejoukkoa

$$\Sigma = \{P(a),\ \forall x(P(x) \to Q(x)),\ \forall x(Q(x) \to Q(f(x)) \land R(x, f(x)))\}.$$

- Lausejoukon Σ Herbrand-universumi on $H = \{f^n(a) \mid n \geq 0\}$.
- Muodostetaan Herbrand-struktuuri on $\mathcal{H}$, jonka universumina on H siten, että jokainen muuttujaton termi $t \in H$ tulkitaan $t^{\mathcal{H}} = t$, ja
$$P^{\mathcal{H}} = \{a\},\ Q^{\mathcal{H}} = H \text{ ja } R^{\mathcal{H}} = \{\langle f^n(a), f^{n+1}(a)\rangle \mid n \geq 0\}.$$
- Kyseinen struktuuri $\mathcal{H}$ on lausejoukon Σ Herbrand-malli.

Määritelmä. Lausejoukon Σ (kielen $\mathcal{L}$) *Herbrand-kanta* B on niiden *atomisten lauseiden* joukko, jotka voidaan muodostaa lausejoukossa Σ esiintyvistä (kielen $\mathcal{L}$) predikaattisymboleista ja vastaavan Herbrand-universumin H muuttujattomista termeistä.

Esimerkki. Edellisen esimerkin tapauksessa Herbrand-kantana on $B = \{P(f^n(a)), Q(f^n(a)) \mid n \geq 0\} \cup \{R(f^n(a), f^m(a)) \mid n \geq 0, m \geq 0\}$.

- Tämä mahdollistaa Herbrand-struktuurien $\mathcal{H}$ määrittelemisen Herbrand-kannan B osajoukkoina: jokaiselle $P \in \mathcal{P}_n$ pätee
$$P(t_1, \ldots, t_n) \in \mathcal{H} \iff \langle t_1, \ldots, t_n\rangle \in P^{\mathcal{H}}.$$
- Herbrand-struktuurille $\mathcal{H}$ voidaan antaa myös literaaliesitys:
$$\begin{aligned} lit(\mathcal{H}) = \{\ & P(a), Q(a), \neg R(a,a), \neg P(f(a)), Q(f(a)), \\ & R(a, f(a)), \neg R(f(a), a), \neg R(f(a), f(a)), \ldots\}. \end{aligned}$$

6.3 Herbrandin teoreema

- Rajoitutaan tarkastelemaan klausuulijoukkoja.
- Merkintä $C(x_1, \ldots, x_n)$ tarkoittaa muuttujat $x_1, \ldots, x_n$ sisältävää klausuulia $\{P_1(\vec{t_1}), \ldots, P_k(\vec{t_k}), \neg Q_1(\vec{s_1}), \ldots, \neg Q_l(\vec{s_l})\}$.
- Kausuuli $C(x_1, \ldots, x_n)$ vastaa universaalisti kvantifioitua lausetta $\forall x_1 \cdots \forall x_n \phi_C(x_1, \ldots, x_n)$, missä $\phi_C(x_1, \ldots, x_n)$ on klausuulin $C(x_1, \ldots, x_n)$ esitys literaalien disjunktiona.
- Klausuulijoukolle S voidaan määritellä Herbrand-struktuurit samaan tapaan kuin lausejoukoillekin.
- Klausuulijoukko S voidaan *instantioida* vastaavan Herbrand-universumin H_S suhteen seuraavasti.

Määritelmä. Klausuulijoukon S *Herbrand-instanssien joukko* S' koostuu muuttujattomista klausuuleista $C(t_1, \ldots, t_n)$, missä $C(x_1, \ldots, x_n) \in S$ ja muuttujattomat termit $t_1 \in H_S, \ldots, t_n \in H_S$.

- Mikäli S on äärellinen sekä H_S on äärellinen (tai vaihtoehtoisesti S on muuttujaton), myös S' on äärellinen.
- Joukko S' voidaan tulkita lauselogiikan klausuulijoukoksi (atomisina lauseina Herbrand-kannan B atomiset lauseet).

Esimerkki. Tarkastellaan klausuulijoukkoa $S = \{\{P(a)\}, \{\neg P(x), P(f(x))\}\}$.

1. Herbrand-universumi $H_S = \{a, f(a), f(f(a)), \ldots\}$.
2. Herbrand-instanssien joukko $S' = \{\{P(a)\}, \{\neg P(a), P(f(a))\}, \{\neg P(f(a)), P(f(f(a)))\}, \ldots\}$.

Väite. (Hebrandin teoreema). Olkoon S joukko klausuuleita ja S' sen Herbrand-instanssien joukko. Tällöin

1. S on toteutumaton $\iff$ S' on toteutumaton, ja
2. S on toteutumaton $\iff$ $\exists$ joukon S' äärellinen osajoukko S'', joka on toteutumaton.

Huomioita. Predikaattilogiikan tapauksessa voidaan täten rajoittua *syntaktisiin malleihin* (Herbrand-malleihin) mielivaltaisten mallien sijaan.

Herbrandin teoreema johtaa myös naiiviin proseduuriin klausuulijoukon S toteutuvuusongelman ratkaisemiseksi:

(i) tuotetaan äärellinen Herbrand-instanssien osajoukko S'' ja

(ii) testataan, onko S'' on toteutumaton. Jos on, lopetetaan ja todetaan S toteutumattomaksi. Muutoin jatketaan kohdasta (i).

6.4 Lauselogiikan ja predikaattilogiikan suhteesta

- Lauselogiikka on osa predikaattilogiikkaa:
  - Kaikki lauselogiikan konnektiivit ovat käytettävissä predikaattilogiikassa.
  - 0-paikkaiset predikaatit vastaavat atomisia lauseita.
- Lauselogiikan päätelmät ja loogiset ongelmat voidaan suorittaa/ratkoa sellaisenaan predikaattilogiikan puitteissa.
- Herbrandin teoreeman nojalla predikaattilogiikan päättely voidaan palauttaa lauselogiikan päättelyksi.
- Lauselogiikan ja predikaattilogiikan *ilmaisuvoimassa* (eli kyvyssä esittää tietämystä) on kuitenkin huomattava ero.

- Ilmaisuvoimaeron ilmentyminen:
  - Äärellistä predikaattilogiikan lausejoukkoa saattaa vastata ääretön lauselogiikan lausejoukko.
  - Lauselogiikan ratkeavuus vs. predikaattilogiikan puoliratkeavuus.
- Rajoittamalla syntaksia sopivasti saadaan predikaattilogiikallekin ratkeavia (ja ilmaisuvoimaltaan heikompia) osajoukkoja.
  - Esim. jos S on äärellinen ja siinä ei esiinny funktiosymboleja, sen Herbrand-instanssien joukko S' jää äärelliseksi.
  - Tällöin S':n toteutuvuus on selvitettävissä äärellisessä ajassa.

Esimerkki. Klausuulijoukon $S = \{\{P(a)\}, \{\neg P(x), P(b)\}\}$ Herbrand-universumi $H = \{a, b\}$ ja Herbrand-instanssien joukko $S' = \{\{P(a)\}, \{\neg P(a), P(b)\}, \{\neg P(b), P(b)\}\}$, joka voidaan nähdä lauselogiikan klausuulijoukkona $S' = \{\{P\}, \{\neg P, Q\}, \{\neg Q, Q\}\}$.

7 Unifikaatio

- Substituutiot
- Yleisimmät unifioijat
- Unifikaatioalgoritmi

7.1 Substituutiot

Määritelmä. *Substituutio* (tai *korvaus*) θ on äärellinen joukko

$$\{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\},$$

missä x_i:t ovat muuttujia ja t_i:t korvaavia termejä siten, että

1. korvattavat muuttujat $x_1, \ldots, x_n$ ovat toisistaan eriävät ja
2. mikään korvaava termi t_i ei ole muuttuja x_i itse eli $t_i \neq x_i$.

Lisäksi erotetaan seuraavat erikoistapaukset:

- Jos korvaavat termit t_i ovat muuttujattomia, θ on *muuttujaton*.
- Jos korvaavat termit t_i ovat muuttujia, θ on *nimeämissubstituutio*.

Esimerkki. Esimerkkeinä todettakoon

- tyhjä substituutio $\epsilon = \{\}$,
- substituutio $\theta_1 = \{x/y, y/a, z/f(w)\}$,
- muuttujaton substituutio $\theta_2 = \{x/a, y/g(c, c)\}$ ja
- nimeämissubstituutio $\theta_3 = \{x/y, y/z, z/x\}$.

Määritelmä. Olkoon E jokin *lauseke* (eli termi, atomikaava, literaali, klausuuli tms.) ja $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$ substituutio.

Lauseke $E\theta$ on muutoin rakenteeltaan kuten E, paitsi että jokainen muuttujan x_i esiintymä lausekkeessa E on korvattu termillä t_i.

Jos lausekkeessa $E\theta$ ei esiinny muuttujia, kutsutaan lauseketta $E\theta$ lausekkeen E *muuttujattomaksi instanssiksi*.

Esimerkki. Olkoon lauseke $E = P(x, y, f(z), v, w)$ ja substituutio $\theta = \{x/y, y/x, z/x, v/f(z), w/g(f(y), c)\}$.

Tällöin $E\theta$ on $P(y, x, f(x), f(z), g(f(y), c))$.

Määritelmä. Olkoot $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$ ja $\lambda = \{y_1/u_1, \ldots, y_m/u_m\}$ kaksi substituutiota.

Substituutioiden θ ja λ *kompositio* $\theta\lambda$ määritellään joukkona

$$\{x_i/t_i\lambda \mid i \in \{1, \ldots, n\} \text{ ja } x_i \neq t_i\lambda\} \cup$$
$$\{y_i/u_i \mid i \in \{1, \ldots, m\} \text{ ja } y_i \notin \{x_1, \ldots, x_n\}\}.$$

Huomio. Määritelmän tavoitteena on saada aikaan kokonaisvaikutus $E(\theta\lambda) = (E\theta)\lambda$ mille tahansa lausekkeelle E.

Esimerkki. Substituutioiden $\theta = \{x/f(y), y/z\}$ ja $\lambda = \{x/a, y/b, z/y\}$. kompositio on $\{x/f(b), z/y\}$.

7.2 Yleisimmät unifioijat

Määritelmä. Olkoon $S = \{E_1, \ldots, E_n\}$ joukko lausekkeita. Substituutio θ on lausekejoukon S *unifioija*, jos $E_1\theta = E_2\theta = \cdots = E_n\theta$.

Lausekejoukko S on *unifioituva*, mikäli sillä on ainakin yksi unifioija.

Esimerkki. Tarkastellaan seuraavien joukkojen unifioituvuutta.

Joukko S:	Unifioija θ:
$\{P(x, f(a)), P(y, z)\}$	$\{y/x, z/f(a)\}$ tai $\{x/y, z/f(a)\}$
$\{P(x, f(x)), P(f(a), y)\}$	$\{x/f(a), y/f(f(a))\}$
$\{P(a), P(f(x))\}$	ei unifioijaa
$\{P(x), P(f(x))\}$	ei unifioijaa (termit aina äärellisiä)

Määritelmä. Olkoon σ lausekejoukon $S = \{E_1, \ldots, E_n\}$ unifioija.

Substituutiota σ kutsutaan lausekejoukon S *yleisimmäksi unifioijaksi*, mikäli jokainen S:n unifioija $\theta = \sigma\lambda$ jollekin substituutiolle λ.

- Vastaava käsite englanniksi on *most general unifier* (MGU).

Esimerkki. Joukon $S = \{P(x, f(y)), P(a, z)\}$ unifioijia ovat mm. $\theta = \{x/a, z/f(b), y/b\}$ ja $\sigma = \{x/a, z/f(y)\}$.

Näistä σ on L:n yleisin unifioija, koska esim. $\theta = \sigma\{y/b\}$.

- Yleisimmät unifioijat ovat yksikäsitteisiä seuraavaan tapaan:

Väite. Olkoot θ ja σ joukon S yleisimpiä unifioijia. Tällöin on olemassa nimeämissubstituutio λ siten että $S\theta\lambda = S\sigma$.

7.3 Unifikaatioalgoritmi

- Tavoitteena laskea atomikaavojen joukolle $S \neq \emptyset$ yleisin unifioija σ.

Määritelmä. Olkoon S ei-tyhjä joukko johonkin predikaattisymboliin P perustuvia atomikaavoja $\{P(\vec{t_1}), \ldots, P(\vec{t_n})\}$.

1. Joukon S *erokohta* on vasemmalta oikealle siirryttäessä ensimmäinen kohta, jossa joukon S atomikaavojen merkkijonoesityksissä on jokin eroavaisuus.
2. Joukon S *erojoukkoon* $D(S)$ kuuluvat atomikaavojen $P(\vec{t_1}), \ldots, P(\vec{t_n})$ erokohdasta alkavat termit $u_1, \ldots, u_n$.

Esimerkki. Joukon $S_1 = \{P(x, a), P(x, y)\}$ erojoukko $D(S_1) = \{a, y\}$. Joukon $S_2 = \{Q(g(x, y), y),\ Q(g(x, f(z)), x),\ Q(g(x, x), f(a))\}$ erojoukko $D(S_2) = \{y, f(z), x\}$.

Unifikaatioalgoritmi ei-tyhjälle atomikaavojen joukolle S:

1. Jos joukon S atomikaavojen predikaattisymbolit eivät ole samat, totea, ettei S unifioidu ja lopeta algoritmin suoritus.
2. Aseta $k := 0$, $S_k := S$ ja $\sigma_k := \epsilon$.
3. Jos S_k on yksialkioinen (ja siten jo unifioitunut) joukko, totea S unifioituvaksi ja lopeta algoritmin suoritus.
4. Laske joukon S_k erojoukko $D(S_k)$.
5. Jos $D(S_k)$:ssa on muuttuja v_k ja termi t_k siten, että v_k ei esiinny t_k:ssa, jatka algoritmin suoritusta kohdasta 7.
6. Muutoin totea, ettei S ole unifioituva, ja lopeta algoritmin suoritus.
7. Aseta $\sigma_{k+1} := \{v_k/t_k\}$ ja laske $S_{k+1} := S_k\{v_k/t_k\}$.
8. Aseta $k := k + 1$ ja jatka algoritmin suorittamista kohdasta 3.

Väite. Olkoon S äärellinen ei-tyhjä joukko atomikaavoja.

- Jos S on unifioituva, niin unifikaatioalgoritmin suoritus päättyy askeleen 3 kohdalla ja substituutioiden $\sigma_0, \sigma_1, \ldots, \sigma_k$ kompositio $\sigma = \sigma_0\sigma_1 \ldots \sigma_k$ on joukon S yleisin unifioija
- Jos S ei ole unifioituva, niin unifikaatioalgoritimin laskenta päättyy askeleessa 1 tai askeleessa 6.

Esimerkki. Lasketaan unifikaatioalgoritmilla joukon $S = \{P(x, f(x)), P(g(a), z)\}$ yleisin unifioija:

1. Predikaattisymbolit ovat samat, jatketaan.
2. $k = 0, S_0 = \{P(x, f(x)), P(g(a), z)\}, \sigma_0 = \epsilon$.
3. S_0 ei ole yksialkoinen, jatketaan.

4. $D(S_0) = \{x, g(a)\}$.
5. Valitaan muuttuja $v_0 = x$ ja termi $t_0 = g(a)$.
7. $\sigma_1 = \{x/g(a)\}, S_1 = \{P(g(a), f(g(a))), P(g(a), z)\}$.
8. $k = 1$.
3. S_1 ei ole yksialkioinen, jatketaan.
4. $D(S_1) = \{f(g(a)), z\}$.
5. Valitaan muuttuja $v_1 = z$ ja termi $t_1 = f(g(a))$.
7. $\sigma_2 = \{z/f(g(a))\}$, $S_2 = \{P(g(a), f(g(a)))\}$.
8. $k = 2$.
3. S_2 on yksialkioinen, joten S on unifioituva.

Yleisin unifioija on $\sigma = \sigma_0\sigma_1\sigma_2 = \epsilon\{x/g(a)\}\{z/f(g(a))\} = \{x/g(a), z/f(g(a))\}$ ja $S\sigma = \{P(g(a), f(g(a)))\} = S_2$.

8 Resoluutiosääntö ja -todistukset

- Resoluutiosääntö predikaattilogiikan tapauksessa
- Resoluutiotodistukset
- Ohjeita resoluutiotodistusten kirjoittamiseen
- Kytkennät logiikkaohjelmointiin (PROLOG)

8.1 Resoluutiosääntö predikaattilogiikan tapauksessa

Määritelmä. Olkoot

$$C_1 = C_1' \sqcup \{P(\vec{t_1}), \ldots, P(\vec{t_n})\} \text{ ja } C_2 = C_2' \sqcup \{\neg P(\vec{u_1}), \ldots, \neg P(\vec{u_m})\}$$

kaksi klausuulia,

1. joissa *ei esiinny yhteisiä* muuttujia ja
2. joissa esiintyvien atomikaavojen joukko
$$\{P(\vec{t_1}), \ldots, P(\vec{t_n}), P(\vec{u_1}), \ldots, P(\vec{u_m})\}$$
on unifioituva (yleisimpänä unifioijana σ).

Klausuulien C_1 ja C_2 *yhdistelmä* on klausuuli $C_1'\sigma \cup C_2'\sigma$.

Huomio. Yllä käytetty merkintä $A \sqcup B$ tarkoittaa keskenään alkiovieraiden ($A \cap B = \emptyset$) joukkojen A ja B unionia $A \cup B$.

Esimerkki. Tarkastellaan seuraavia klausuuleja:

$$C_1 = \{Q(x), \neg R(y), P(x,y), P(f(z), f(z))\} \text{ ja}$$
$$C_2 = \{\neg N(u), \neg R(w), \neg P(f(a), f(a)), \neg P(f(w), f(w))\}.$$

Klausuuleissa ei esiinny yhteisiä muuttujia ja joukon

$$\{P(x,y), P(f(z), f(z)), P(f(a), f(a)), P(f(w), f(w))\}$$

yleisin unifioija on $\sigma = \{x/f(a), y/f(a), z/a, w/a\}$. Klausuulien yhdistelmäksi saadaan $\{Q(f(a)), \neg R(f(a)), \neg N(u), \neg R(a)\}$.

Esimerkki. (Faktorointi) Klausuleilla voi olla useita eri yhdistelmiä. Klausulien $\{P(x_1), P(y_1)\}$ ja $\{\neg P(x_2), \neg P(y_2)\}$ yhdistelmiä ovat mm.

- $\{P(x_1), \neg P(x_2)\}$ (joukolle $\{P(y_1), P(y_2)\}$ MGU $\sigma = \{y_2/y_1\}$) ja
- tyhjä klausuuli $\Box$ (joukolle $\{P(x_1), P(y_1), P(x_2), P(y_2)\}$ MGU $\sigma = \{y_1/x_1, x_2/x_1, y_2/x_1\}$).

8.2 Resoluutiotodistukset

- Lähtökohtana on joukko klausuuleita S, jonka klausuuleista johdetaan uusia klausuuleita resoluutiosäännöllä.
- *Johtojen* ja *hylkäyksen* määritelmät säilyvät ennallaan, mutta resoluutioaskeleiden tulee täyttää resoluutiosäännön vaatimukset.
- Tarvittaessa klausuulien muuttujat tulee nimetä uudelleen.
- Resoluutio on myös predikaattilogiikan tapauksessa virheetön ja täydellinen menettely klausuulijoukon toteutuvuuden tutkimiseen.

Väite. Klausuulijoukolle S löytyy hylkäys (eli klausuulijoukosta S on johto $C_1, \ldots, C_n$ tyhjälle klausuulille $C_n = \Box$) $\iff$ S on toteutumaton.

Todistus. Sivuutetaan.

Esimerkki. Osoitetaan predikaattilogiikan lausejoukko

$$\Sigma = \{\forall x \exists y (P(x) \land P(y)),\ \forall x \forall y (P(x) \rightarrow \neg P(y))\}$$

toteutumattomaksi. Haetaan lauseille ensin klausuuliesitykset:

- $\forall x \exists y (P(x) \land P(y)) \rightsquigarrow \forall x (P(x) \land P(f(x))) \rightsquigarrow$ $S_1 = \{\{P(x)\}, \{P(f(x))\}\}$.
- $\forall x \forall y (P(x) \rightarrow \neg P(y))\} \rightsquigarrow \forall x \forall y (\neg P(x) \lor \neg P(y))\} \rightsquigarrow$ $S_2 = \{\{\neg P(x), \neg P(y)\}\}$.

Hylkäys:	1. $\{P(x)\}$	S_1
	2. $\{\neg P(z), \neg P(y)\}$	$S_2\{x/z\}$
	3. $\Box$	1,2,MGU $\{x/y, z/y\}$

$\implies$ $S_1 \cup S_2$ on toteutumaton $\implies$ Σ on toteutumaton.

Muiden loogisten ongelmien ratkominen

- Resoluutiolla voidaan selvittää lauseiden pätevyyttä ja loogista ekvivalenssia sekä tutkia lausejoukon loogisia seuraavuuksia.
- Koska Skolemointi ei säilytä loogista ekvivalenssia vaan toteutuvuuden, nämä tulee muuntaa toteutuvuusongelmiksi.

Väite. Olkoon ϕ ja ψ lauseita ja Σ lausejoukko.

1. Pätevyys: $\models \phi \iff \mathrm{KM}(\{\neg\phi\})$:lle löytyy hylkäys.
2. Ekvivalenssi: $\phi \equiv \psi \iff \mathrm{KM}(\{\neg(\phi \leftrightarrow \psi)\})$:lle löytyy hylkäys.
3. Looginen seuraavuus: $\Sigma \models \phi$
   $\iff$ klausuulijoukolle $\mathrm{KM}(\Sigma \cup \{\neg\phi\})$ löytyy hylkäys.

Yllä $\mathrm{KM}(\Gamma)$ tarkoitaa lausejoukon Γ klausuulimuotoa, mikä saadaan ottamalla yksittäisten lauseiden $\gamma \in \Gamma$ klausuulimuotojen unioni.

Esimerkki. Osoitetaan resoluutiolla, että $\models \forall x P(x) \rightarrow \exists x P(x)$.

- Haetaan lauseen negaatiolle klausuulimuoto:

$$\begin{aligned}
\neg(\forall x P(x) \rightarrow \exists x P(x)) &\rightsquigarrow \forall x P(x) \land \neg\exists x P(x)\\
&\rightsquigarrow \forall x P(x) \land \forall x \neg P(x)\\
&\rightsquigarrow \forall x \forall y (P(x) \land \neg P(y)).\\
&\rightsquigarrow S = \{\{P(x)\}, \{\neg P(y)\}\}.
\end{aligned}$$

- Klausuuleista $\{P(x)\}$ ja $\{\neg P(y)\}$ saadaan tyhjä klausuuli $\Box$ (MGU $\{x/y\}$) yhdellä resoluutioaskelella.
- Täten klausuulijoukolle S on hylkäys
  $\Longrightarrow$ S on toteutumaton
  $\Longrightarrow$ $\neg(\forall x P(x) \rightarrow \exists x P(x))$ on toteutumaton
  $\Longrightarrow$ $\forall x P(x) \rightarrow \exists x P(x)$ on pätevä.

Esimerkki. Osoitetaan lause $\exists x(E(x) \land K(x))$ lausejoukon

$$\Sigma = \{\forall x(I(x) \rightarrow E(x)), \exists x(I(x) \land K(x))\}$$

loogiseksi seuraukseksi. Haetaan tarvittavat klausuulimuodot:

$$\begin{aligned}
\forall x(I(x) \rightarrow E(x)) &\rightsquigarrow \forall x(\neg I(x) \lor E(x))\\
&\rightsquigarrow S_1 = \{\{\neg I(x), E(x)\}\}.\\
\exists x(I(x) \land K(x)) &\rightsquigarrow I(c) \land K(c)\\
&\rightsquigarrow S_2 = \{\{I(c)\}, \{K(c)\}\}\\
\neg\exists x(E(x) \land K(x)) &\rightsquigarrow \forall x \neg(E(x) \land K(x))\\
&\rightsquigarrow \forall x(\neg E(x) \lor \neg K(x))\\
&\rightsquigarrow S_3 = \{\{\neg E(x), \neg K(x)\}\}
\end{aligned}$$

Kokonaisuutena saadaan siis klausuulijoukko $S = S_1 \cup S_2 \cup S_3$ $= \{\{\neg I(x), E(x)\}, \{I(c)\}, \{K(c)\}, \{\neg E(x), \neg K(x)\}\}$.

Hylkäys löydetään esimerkiksi seuraavasti:

1. $\{\neg I(x), E(x)\}$	S
2. $\{I(c)\}$	S
3. $\{K(c)\}$	S
4. $\{\neg E(y), \neg K(y)\}$	$S\{x/y\}$
5. $\{\neg I(y), \neg K(y)\}$	1,4,MGU $\{x/y\}$
6. $\{\neg K(c)\}$	2,5,MGU $\{y/c\}$
7. $\Box$	3,6,MGU ϵ

- Yleisimpien unifioijien kompositio $\{x/y\}\{y/c\}\epsilon = \{x/c, y/c\}$.
- Täten S on toteutumaton
  $\Longrightarrow$ $\Sigma \cup \{\neg\exists x(E(x) \land K(x))\}$ on toteutumaton
  $\Longrightarrow$ $\exists x(E(x) \land K(x))$ on joukon Σ looginen seuraus.

8.3 Ohjeita resoluutiotodistusten kirjoittamiseen

- Muuttujien uudelleennimeäminen on hyvä suorittaa systemaattisesti (esimerkiksi alaindeksien avulla).
- Yksittäistä klausuulijoukon klausuulia saatetaan tarvita useita kertoja resoluutiotodistuksessa (jolloin muuttujien uudelleennimeäminen on välttämätöntä).
- Kirjoita yleisimmät unifioijat (MGU:t) näkyviin.
- Ellet kirjoita todistusta binääripuun muotoon, numeroi klausuulit ja ilmoita, mistä klausuuleista mikin klausuuli on johdettu.
- Laske yleisimpien unifioijien kompositio selvittääksesi kyselyssä esiintyville muuttujille arvot.

Esimerkki. Esitetään listat vakion e (tyhjä lista) ja kaksipaikkaisen funktiosymbolin c avulla (näin lista $[1,2]$ saa esityksen $c(1,c(2,e))$).

Määritellään seuraavat listoja koskevat predikaatit

- $K(x,y) =$
  "listan x alkioina ovat listan y alkiot käänteisessä järjestyksessä":
  1. $K(e,e)$ ja
  2. $\forall x \forall y \forall z \forall v (K(x,y) \wedge L(y,v,z) \rightarrow K(c(v,x),z))$.
- $L(y,v,z) =$ "lista z on lista y, jonka perään on liitetty alkio v":
  3. $\forall x L(e,x,c(x,e))$ ja
  4. $\forall y \forall v \forall z \forall x (L(y,v,z) \rightarrow L(c(x,y),v,c(x,z)))$.

- Johdetaan lauseille klausuulimuodot:

(1) $\rightsquigarrow \{\{K(e,e)\}\}$

(2) $\rightsquigarrow \forall x \forall y \forall z \forall v (\neg K(x,y) \vee \neg L(y,v,z) \vee K(c(v,x),z))$
$\rightsquigarrow \{\{\neg K(x,y), \neg L(y,v,z), K(c(v,x),z)\}\}$

(3) $\rightsquigarrow \{\{L(e,x,c(x,e))\}\}$

(4) $\rightsquigarrow \forall y \forall v \forall z \forall x (\neg L(y,v,z) \vee L(c(x,y),v,c(x,z))$
$\rightsquigarrow \{\{\neg L(y,v,z), L(c(x,y),v,c(x,z))\}\}$

(5) Todistettavan lauseen negaatio $\neg \exists x K(c(1,c(2,e)),x)$
$\rightsquigarrow \forall x \neg K(c(1,c(2,e)),x) \rightsquigarrow \{\{\neg K(c(1,c(2,e)),x)\}\}$

- Haluamme siis selvittää, millainen on lista $[1,2]$ käännettynä.

Resoluutiotodistus:

1. $\{\neg K(c(1,c(2,e)),x_0)\}$ P5
2. $\{\neg K(x_1,y_1), \neg L(y_1,v_1,z_1), K(c(v_1,x_1),z_1)\}$ P2
3. $\{\neg K(c(2,e),y_1), \neg L(y_1,1,z_1)\}$ 1,2,MGU $\{v_1/1, x_1/c(2,e), x_0/z_1\}$
4. $\{\neg K(x_2,y_2), \neg L(y_2,v_2,z_2), K(c(v_2,x_2),z_2)\}$ P2
5. $\{\neg K(e,y_2), \neg L(y_2,2,y_1), \neg L(y_1,1,z_1)\}$
   3,4,MGU $\{v_2/2, x_2/e, z_2/y_1\}$
6. $\{K(e,e)\}$ P1
7. $\{\neg L(e,2,y_1), \neg L(y_1,1,z_1)\}$ 5,6,MGU $\{y_2/e\}$
8. $\{L(e,x_3,c(x_3,e))\}$ P3

9. $\{\neg L(c(2,e),1,z_1)\}$ 7,8,MGU $\{x_3/2, y_1/c(2,e)\}$

10. $\{\neg L(y_4,v_4,z_4), L(c(x_4,y_4),v_4,c(x_4,z_4))\}$ P4

11. $\{\neg L(e,1,z_4)\}$ 9,10,MGU $\{x_4/2, y_4/e, v_4/1, z_1/c(2,z_4)\}$

12. $\{L(e,x_5,c(x_5,e))\}$ P3

13. $\square$ 11,12,MGU $\{x_5/1, z_4/c(1,e)\}$

- Unifioijien kompositio: $\{\ v_1/1,\ x_1/c(2,e),\ x_0/c(2,c(1,e)),$
  $v_2/2,\ x_2/e,\ z_2/c(2,e),$
  $y_2/e,$
  $x_3/2,\ y_1/c(2,e)$
  $x_4/2,\ y_4/e,\ v_4/1,\ z_1/c(2,c(1,e))$
  $x_5/1,\ z_4/c(1,e)\ \}.$
- Rajaus kyselyyn: $\{x_0/c(2,c(1,e))\}$ (ns. *vastaussubstituutio*).

8.4 Kytkennät logiikkaohjelmointiin (PROLOG)

Laskenta perustuu *järjestettyjen klausuulien* väliseen resoluutioon.

Määritelmä. Olkoon $G = \{\neg B_1(\vec{u_1}), \ldots, \neg B_m(\vec{u_m})\}$ kyselyn negaatiota vastaava järjestetty *maaliklausuuli* ja $C = \{A(\vec{t}), \neg A_1(\vec{t_1}), \ldots, \neg A_n(\vec{t_n})\}$ järjestetty *ohjelmaklausuuli* (ohjelman sääntö) siten, että

1. klausuuleilla G ja C ei ole yhteisiä muuttujia ja
2. atomilla $A(\vec{t})$ ja *valintafunktion* R määräämässä literaalissa $R(G) = \neg B_i(\vec{u_i})$ esiintyvällä atomilla $B_i(\vec{u_i})$ on yleisin unifioija θ.

Klausuulien G ja C yhdistelmäksi saadaan järjestetty maaliklausuuli

$$G' = \{\ \neg B_1(\vec{u_1}), \ldots, \neg B_{i-1}(\vec{u_{i-1}}), \neg A_1(\vec{t_1}), \ldots, \neg A_n(\vec{t_n}),\\ \neg B_{i+1}(\vec{u_{i+1}}), \ldots, \neg B_m(\vec{u_m})\ \}\theta.$$

Joitain PROLOGin erityispiirteitä

- Tyypillisessä PROLOG-toteutuksessa muuttujasymbolit erotetaan muista symboleista ison alkukirjamen perusteella.
- Literaalijoukkojen sijaan järjestetyt ohjelma- ja maaliklausuulit kirjoitetaan *sääntöinä* seuraavaan tapaan:

$$\{N(0)\} \rightsquigarrow \texttt{n(0).}$$
$$\{N(s(x)), \neg N(x)\} \rightsquigarrow \texttt{n(s(X)) :- n(X).}$$
$$\{\neg N(s(0)), \neg N(s(s(y)))\} \rightsquigarrow \texttt{:- n(s(0)), n(s(s(Y))).}$$

- Tyypillinen valintafunktio valitsee maaliklausuulin 1. atomin.

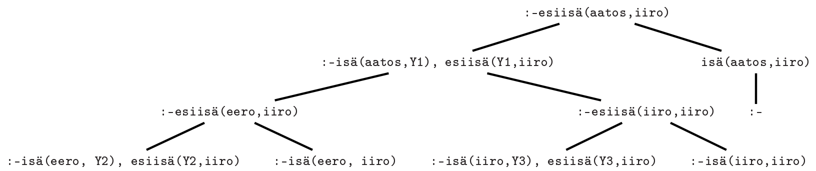
Esimerkki. Tyypillinen PROLOG-toteutus johtaa seuraavat maaliklausuulit: (1) `:- n(0), n(s(s(Y)))`, (2) `:- n(s(s(Y)))`, (3) `:- n(s(Y))`, (4) `:- n(Y)` ja (5) `:-` (tyhjä klausuuli).

Esimerkki. Tarkastellaan seuraavaa PROLOG-ohjelmaa:

```
isä(aatos,eero). isä(aatos,iiro). isä(oiva,aatos).

esiisä(X,Z) :- isä(X,Y), esiisä(Y,Z).
esiisä(X,Y) :- isä(X,Y).
```

Maaliklausuuli `:- esiisä(aatos, iiro)` johtaa seuraavaan hakuun:



$\Longrightarrow$ Koska tyhjän klausuulin `:-` johtaminen onnistuu, PROLOG-tulkki vastaa kyselyyn myöntävästi: `esiisä(aatos, iiro)` on johdettavissa.

9 Ohjelmien oikeellisuustarkastelut

- Tarkasteltava ohjelmointikieli
- Boolen lausekkeiden ekvivalenssi
- Ohjelmien esi- ja jälkiehdot
- Toistolauseiden invariantit
- Täysi oikeellisuus

Motivaatio

Miksi tietokoneohjelmille tulisi kirjoittaa formaaleja spesifikaatioita?

- Spesifikaatiota laadittaessa joudutaan suunnittelemaan ennalta varsin tarkkaan mitä ohjelmiston on tarkoitus tehdä.
- Järjestelmän toteutus voidaan *verifioida* eli todeta määrittelynsä mukaiseksi vasta, kun spesifikaatio on tehty.
- Formaalissa spesifioinnissa etuna on määritelmien yksikäsitteisyys.
- *Turvallisuuskriittiset järjestelmät* (esim. lentokoneen ohjausjärjestelmä) vaativat perinpohjaista määrittelyä ja verifiointia.
- Hyvin määritellyn ohjelman uudelleenkäyttö on helpompaa.

9.1 Tarkasteltava ohjelmointikieli

Tarkastellaan seuraavaa kokonaislukujen käsittelyyn riittävää osajoukkoa tyypillisistä lausekielistä kuten Pascal, C, C++ ja Java.

Määritelmä. *Kokonaislukulausekkeet* E määritellään seuraavasti.

1. Mikä tahansa kokonaisluku ..., `-1`, `0`, `1`, ... on kokonaislukulauseke.
2. Kokonaislukumuuttujat `x`, `y`, ... ovat kokonaislukulausekkeita.
3. Jos E_1 ja E_2 ovat kokonaislukulausekkeita, niin myös

   summa $(E_1 + E_2)$, erotus $(E_1 - E_2)$ ja tulo $(E_1 * E_2)$

   ovat kokonaislukulausekkeita.
4. Muita kokonaislukulausekkeita ei ole.

Esimerkki. Merkkijono `((x - y) * x)` on kokonaislukulauseke.

Määritelmä. *Boolen lausekkeet* B määritellään seuraavasti.

1. Boolen vakiot `false` ja `true` ovat Boolen lausekkeita.
2. Jos E_1 ja E_2 ovat kokonaislukulausekkeita, niin $E_1 > E_2$ on Boolen lauseke.
3. Jos B_1 ja B_2 ovat Boolen lausekkeita, niin

   negaatio $!B_1$, konjunktio B_1 && B_2 ja disjunktio $B_1 \,||\, B_2$

   ovat myös Boolen lausekkeita.
4. Muita Boolen lausekkeita ei ole.

Huomioita. Lausekkeet $E_1 == E_2$ (yhtäsuuruus), $E_1 \mathrel{!=} E_2$, $E_1 <= E_2$, $E_1 < E_2$ ja $E_1 >= E_2$ ovat lyhennysmerkintöjä lausekkeille $!(E_1 > E_2)$ && $!(E_2 > E_1)$, $!(E_1 == E_2)$, $!(E_1 > E_2)$, $E_2 > E_1$ ja $!(E_1 < E_2)$; sekä implikaatio $B_1 \mathrel{->} B_2$ lausekkeelle $!B_1 \,||\, B_2$.

Määritelmä. Ko. ohjelmointikielen *komennot* C ovat seuraavat:

1. Jos x on kokonaislukumuuttuja ja E on kokonaislukulauseke, niin *sijoituslause* `x = `E on komento.
2. Jos B on Boolen lauseke sekä C_1 ja C_2 ovat komentoja, niin myös

C_1 ; C_2	(ketjulause)
`if` (B) `then` $\{C_1\}$ `else` $\{C_2\}$	(ehtolause)
`while` (B) $\{C_1\}$	(toistolause)

   ovat komentoja.
3. Muita komentoja ei ole.

$\Longrightarrow$ *Ohjelmat* ovat määritelmän mukaisia (rakenteisia) komentoja.

Esimerkki. Ohjelma `y = 1 ; z = 1 ; while(z != x) {z = z + 1 ; y = y * z}` laskee muuttujan `y` arvoksi muuttujan arvon `x` kertoman (kun `x > 0`).

Ohjelman tilojen esittäminen struktuurina

Määritelmä. Struktuuri $\mathcal{S}$ on $\mathbb{Z}$-struktuuri seuraavilla edellytyksillä:

(i) Struktuurin $\mathcal{S}$ universumina U on kokonaislukujen joukko $\mathbb{Z}$.

(ii) Jokaisen kokonaisluvun (vakiosymboli tarkasteltavassa kielessä) tulkintana on kyseinen kokonaisluku itse.

(iii) Funktiosymbolien `+`, `-` ja `*` tulkintoina ovat yhteen-, vähennys- ja kertolaskufunktiot kokonaislukujen joukossa.

(iv) Predikaattisymbolin `>` tulkintana on suurempi kuin -relaatio kokonaislukujen joukossa.

Huomioita.

- Ohjelman suorituksen *tila* voidaan rinnastaa $\mathbb{Z}$-struktuuriin $\mathcal{S}$.
- Kokonaislukulausekkeen E arvo tilassa $\mathcal{S}$ on kokonaisluku $E^{\mathcal{S}}$.
- Boolen lauseke B on tosi tilassa $\mathcal{S}$ $\iff$ $\mathcal{S} \models B$.

Esimerkki. Tarkastellaan tilaa $\mathcal{S}$, missä $\mathtt{x}^{\mathcal{S}} = 2$ ja $\mathtt{y}^{\mathcal{S}} = 6$ ja $\mathtt{z}^{\mathcal{S}} = 3$.

Lausekkeiden $(\mathtt{x} * \mathtt{x})$ ja $(\mathtt{z} * \mathtt{z})$ arvot ovat $(\mathtt{x} * \mathtt{x})^{\mathcal{S}} = 4$ ja $(\mathtt{z} * \mathtt{z})^{\mathcal{S}} = 9$.

Niinpä $\mathcal{S} \models (\mathtt{x} * \mathtt{x} < \mathtt{y})\ \&\&\ (\mathtt{y} < \mathtt{z} * \mathtt{z})$, mutta $\mathcal{S} \not\models (\mathtt{x} * \mathtt{y} < \mathtt{z})$.

Komentojen suorittamisen vaikutus tilaan

Määritelmä. Määritellään tilansiirtorelaatio $\mathcal{S} \xrightarrow{C} \mathcal{S}'$ määräämällä tila $\mathcal{S}'$, johon päädytään tilasta $\mathcal{S}$, jos ja kun komennon C suoritus päättyy.

1. Jos C on sijoituslause $\mathtt{x} = E$, niin tila $\mathcal{S}'$ on $\mathcal{S}[\mathtt{x} \mapsto E^{\mathcal{S}}]$.
2. Jos C on ketjulause C_1 ; C_2, niin $\mathcal{S}'$ on tila, joka saavutetaan suorittamalla ensin C_1 ja sitten C_2.
3. Jos C on ehtolause `if` (B) `then` $\{C_1\}$ `else` $\{C_2\}$, niin $\mathcal{S}'$ on tila, joka saavutetaan suorittamalla C_1, jos $\mathcal{S} \models B$, ja C_2, jos $\mathcal{S} \not\models B$.
4. Jos C on toistolause `while` (B) $\{C_1\}$ ja $\mathcal{S} \not\models B$, niin $\mathcal{S}' = \mathcal{S}$.
5. Jos C on toistolause `while` (B) $\{C_1\}$ ja $\mathcal{S} \models B$, niin $\mathcal{S}'$ on tila, joka saavutetaan suorittamalla C_1 ; `while` (B) $\{C_1\}$.

Huomioita.

- Ainoastaan sijoituslauseiden suorittaminen voi muuttaa tilaa.
- `while`-rakenteen suorittamisen päättyminen ei ole taattua.

Esimerkki. Tarkastellaan ohjelman

```
y = 1 ; z = 1 ; while(z != x) {z = z + 1 ; y = y * z}
```

suoritusta tilasta $\mathcal{S}$, missä $\mathtt{x}^{\mathcal{S}} = 3$. Ohjelman suorituksen aikana alkutilaa päivitetään seuraavasti:

$$\mathtt{y} \mapsto 1,\ \mathtt{z} \mapsto 1,\ \mathtt{z} \mapsto 2,\ \mathtt{y} \mapsto 2,\ \mathtt{z} \mapsto 3 \text{ ja } \mathtt{y} \mapsto 6.$$

$\Longrightarrow$ muuttujan `y` arvona on muuttujan `x` arvon kertoma.

Esimerkki. Ohjelman `x = 1 ; while(x > 0) {y = y + 1}` suoritus ei pääty, koska komennon `y = y + 1` toistaminen ei vaikuta ehdon toteutumiseen.

9.2 Boolen lausekkeiden ekvivalenssi

- Ohjelmointikielissä käytetään paljon ehtolauseita kontrolloimaan, millä ehdoilla ja mitä toimintoja suoritetaan.
- Jos ehtolauseita muutetaan esim. optimointitarkoituksessa, halutaan varmistua että toiminnot suoritetaan samoilla ehdoilla.
- Boolen lausekkeiden ekvivalenssin osoittamiseen voidaan käyttää sekä lauselogiikan että predikaattilogiikan menetelmiä.
- Jos Boolen lausekkeiden evaluoinnilla on *sivuvaikutuksena* muutoksia ohjelman tilaan, pelkkä loogisen ekvivalenssin tarkastaminen ei välttämättä riitä.

Esimerkki. Tällainen sivuvaikutus voi olla esim. virhetilanne, joka on aiheutunut alilausekkeiden evaluoinnista väärässä järjestyksessä.

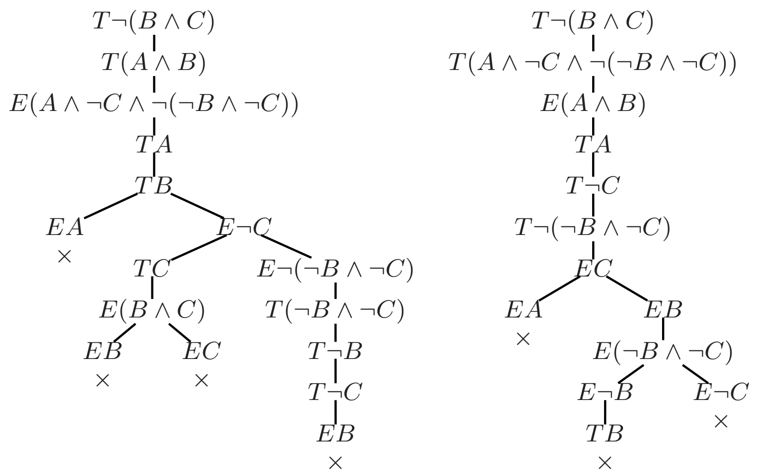
Esimerkki. Vertaillaan kahta eri ohjelmaa:

```
if((x > 0) && !(y > x)) then {
    if(x != y) then {z = x} else {z = y}
} else {z = 0}
```

```
if(x > 0) then {
    if(x > y) then {z = x} else {z = y}
} else {z = 0}
```

- Valitaan atomiset lauseet A = "`x > 0`", B = "`x > y`" ja C = "`y > x`".
- Nyt esim. sijoituslause `z = x` suoritetaan näissä ohjelmissa seuraavilla ehdoilla: $A \wedge \neg C \wedge \neg(\neg B \wedge \neg C)$ ja $A \wedge B$.
- Lauselogiikan nojalla näiden välinen ekvivalenssi on looginen seuraus lauseesta $\neg(B \wedge C)$, joka on aina voimassa B:n ja C:n välillä.

Esimerkki. (Jatkoa) ekvivalenssi voidaan todeta semanttisella taululla:



Näiden perusteella $\{\neg(B \wedge C)\} \models (A \wedge B) \leftrightarrow (A \wedge \neg C \wedge \neg(\neg B \wedge \neg C))$.

Kytkennät predikaattilogiikkaan

- Boolen lauseke B on $\mathbb{Z}$-*pätevä* (merk. $\models_{\mathbb{Z}} B$) $\iff$ $\mathcal{S} \models B$ kaikissa $\mathbb{Z}$-struktuureissa $\mathcal{S}$.
- Näin lausekkeiden muuttujat saavat universaalin tulkinnan.
- Boolen lausekkeet B_1 ja B_2 ovat $\mathbb{Z}$-*ekvivalentit* (merk. $B_1 \equiv_{\mathbb{Z}} B_2$) $\iff$ lausekkeilla on sama totuusarvo kaikissa $\mathbb{Z}$-struktuureissa.
- Huomaa, että $\models B \implies \models_{\mathbb{Z}} B$ ja $B_1 \equiv B_2 \implies B_1 \equiv_{\mathbb{Z}} B_2$, mutta käänteiset implikaatiot eivät välttämättä ole voimassa.

Esimerkki. $\models_{\mathbb{Z}} \mathtt{!((x>y)\,\&\&\,(y>x))}$, mutta $\not\models \mathtt{!((x>y)\,\&\&\,(y>x))}$, koska löytyy vastamalli $\mathcal{S}$, missä $U = \{0\}$, $\mathtt{x}^{\mathcal{S}} = \mathtt{y}^{\mathcal{S}} = 0$ ja $>^{\mathcal{S}} = \{\langle 0, 0\rangle\}$.

$\implies$ relaation > suhde itseensä ja funktioihin +, - ja * on kuvattava erikseen (vrt. $\neg(B \wedge C)$ edellä), jos käytetään predikaattilogiikkaa.

9.3 Ohjelmien esi- ja jälkiehdot

- Tarkasteltavan ohjelmointikielen ohjelmilla on ääretön tila-avaruus, jonka läpikäyminen on käytännössä mahdotonta.
- Yksi mahdollisuus on tarkastella Boolen lausekkeiden B määrittelemiä tilajoukkoja $\{\mathcal{S} \mid \mathcal{S} \models B\}$ ja analysoida, millaisia muutoksia annettu ohjelma näihin aiheuttaa.
- Mille tahansa ohjelmalle P voidaan asettaa *esi- ja jälkiehdot* B_1 ja B_2 kirjoittamalla ns. Hoaren kolmikko $[B_1]\ \mathtt{P}\ [B_2]$.
- Karkeasti ottaen ajatuksena on, että esiehdon B_1 on tarkoitus taata jälkiehdon B_2 voimaantulo ohjelman P suorituksen päättyessä.

Esimerkki. Ohjelmalle Succ: $\mathtt{if\,(x==0)\ then\,\{y=1\}\ else\,\{y=x+1\}}$ voidaan antaa spesifikaatio $[\mathtt{true}]\ \mathtt{Succ}\ [\mathtt{y==x+1}]$.

Osittainen ja täysi oikeellisuus

Olkoon P ohjelma sekä B_1 ja B_2 kaksi Boolen lauseketta.

Määritelmä. Ohjelma P on *osittain oikeellinen* annettujen esi- ja jälkiehtojen B_1 ja B_2 suhteen (merk. $\models_p [B_1]\ \mathtt{P}\ [B_2]$) $\iff$ $\mathcal{S}' \models B_2$ pätee saavutettavalle tilalle $\mathcal{S}'$ aina kun ohjelman P suoritus aloitetaan tilasta $\mathcal{S}$, missä $\mathcal{S} \models B_1$, ja ohjelman P *suoritus päättyy* tilaan $\mathcal{S}'$.

Esimerkki. Osittainen oikeellisuus ei edellytä suorituksen päättymistä: $\models_p [\mathtt{true}]\ \mathtt{while\,(x\,!=y)\,\{z=x\,;\,x=y\,;\,y=z\}}\ [\mathtt{x==y}]$.

Määritelmä. Ohjelma P on *täysin oikeellinen* suhteessa annettuihin esi- ja jälkiehtohin B_1 ja B_2 (merk. $\models_t [B_1]\ \mathtt{P}\ [B_2]$) $\iff$ $\models_p [B_1]\ \mathtt{P}\ [B_2]$ ja ohjelman P suoritus päättyy aina kun $\mathcal{S} \models B_1$ alkutilalle $\mathcal{S}$.

Huomio. Vastaavat englannin kieliset termit ovat *partial correctness* ($\models_p$) ja *total correctness* ($\models_t$).

Päättelysääntöjä osittaiselle oikeellisuudelle

Boolen lausekkeille B, B_0, B_1 ja B_2 sekä komennoille C, C_1 ja C_2:

Sijoituslause: $$\frac{}{[B\{\mathtt{x}/E\}]\ \mathtt{x}=E\ [B]}$$

Kompositio: $$\frac{[B_0]\ C_1\ [B_1] \qquad [B_1]\ C_2\ [B_2]}{[B_0]\ C_1\ \mathtt{;}\ C_2\ [B_2]}$$

Ehtolause: $$\frac{[B_1\ \mathtt{\&\&}\ B]\ C_1\ [B_2] \qquad [B_1\ \mathtt{\&\&}\ \mathtt{!}B]\ C_2\ [B_2]}{[B_1]\ \mathtt{if}\,(B)\ \mathtt{then}\,\{C_1\}\ \mathtt{else}\,\{C_2\}\ [B_2]}$$

Toistolause: $$\frac{[B_1\ \mathtt{\&\&}\ B_2]\ C\ [B_1]}{[B_1]\ \mathtt{while}\,(B_2)\,\{C\}\ [B_1\ \mathtt{\&\&}\ \mathtt{!}B_2]}$$

Implikaatio: $$\frac{\models_{\mathbb{Z}} B_1\ \mathtt{->}\ B_2 \qquad [B_2]\ C\ [B_3] \qquad \models_{\mathbb{Z}} B_3\ \mathtt{->}\ B_4}{[B_1]\ C\ [B_4]}$$

Esimerkki. Osoitetaan
$\models_p$ `[(x == n) && (y == m)] z = x ; x = y ; y = z [(x == m) && (y == n)]`.

Todistus. Käytetään edellä esiteltyjä päättelysääntöjä:

1.	`[(x == m) && (z == n)] y = z [(x == m) && (y == n)]`	Sij.
2.	`[(y == m) && (z == n)] x = y [(x == m) && (z == n)]`	Sij.
3.	`[(y == m) && (z == n)] x = y ; y = z [(x == m) && (y == n)]`	1,2,Komp.
4.	`[(y == m) && (x == n)] z = x [(y == m) && (z == n)]`	Sij.
5.	`[(y == m) && (x == n)] z = x ; x = y ; y = z [(x == m) && (y == n)]`	3,4,Komp.
6.	`[(x == n) && (y == m)] z = x ; x = y ; y = z [(x == m) && (y == n)]`	Impl.

Viimeisessä askelessa hyödynnetään Boolen lausekkeiden `(x == n) && (y == m)` ja `(y == m) && (x == n)` välistä ekvivalenssia.

Huomioita. Todistus voitaisiin kirjoittaa myös puun muotoon.

Heikoimmat esiehdot

- Olkoon `P` ohjelma C_1 ; ... ; C_n, missä $C_1, \ldots, C_n$ ovat järjestyksessä peräkkäin suoritettavat komennot.
- Ominaisuuden $\models_p [B_0]$ `P` $[B_n]$ osoittaminen voidaan pilkkoa osaongelmiin: tulisi löytää sopivat ehdot $B_1, \ldots, B_{n-1}$ siten, että $\models_p [B_{i-1}]\ C_i\ [B_i]$ on osoitettavissa kaikille $i \in \{1, \ldots, n\}$.
- Usein tällaiset ehdot voidaan löytää analysoimalla komentosekvenssiä takaperin: haetaan komennolle C_i (missä i saa arvot n, $n-1$, ..., 1) *heikoin esiehto* B_{i-1} siten, että $\models_p [B_{i-1}]\ C_i\ [B_i]$.
- Jatkossa tällaisia todistuksia kirjoitetaan sekvensseiksi
$$[B_0]\ C_1\ [B_1]\ C_2\ [B_2] \ldots\ C_n\ [B_n],$$
vaikka käytännössä sekvenssi muodostetaankin usein takaperin.

Tarkastellaan seuraavaksi, millaisista todistusaskelista tällainen sekvenssi voidaan muodostaa edellä esiteltyjen päättelysääntöjen nojalla.

1. Jos B on jälkiehto sijoituslauseelle `x` $= E$, heikoimmaksi esiehdoksi voidaan kirjata $B\{\mathtt{x}/E\}$ eli $\models_p [B\{\mathtt{x}/E\}]\ \mathtt{x} = E\ [B]$.
   Esimerkki. `[x - 1 > 0] y = x - 1 [y > 0]`
2. Jos $\models_p [B_1]\ C\ [B_2]$ on jo osoitettu ja B_0 on esiehdon B_1 vahvennus ($\models_{\mathbb{Z}} B_0$ `->` B_1), kirjataan $[B_0]\ [B_1]\ C\ [B_2]$, koska $\models_p [B_0]\ C\ [B_2]$.
   Esimerkki. `[x > 1] [x - 1 > 0] y = x - 1 [y > 0]`
3. Jos $\models_p [B_1]\ C_1\ [B_3]$ ja $\models_p [B_2]\ C_2\ [B_3]$ ovat jo osoitetut jälkiehdolle B_3, niin ehtolauseen `if` (B) `then` $\{C_1\}$ `else` $\{C_2\}$ heikoimmaksi esiehdoksi kirjataan $(B$ `&&` $B_1)$ `||` $($`!`B `&&` $B_2)$.
   Esimerkki. `[!(x > y)] [((x > y) && (x == y)) || (!(x > y) && (y == y))]`
   `if (x > y) then {x = x} else {x = y} [x == y]`

Esimerkki. Tarkastellaan ohjelman `Succ` muunnelmaa.

```
[true]
[(x == 0) || !(x == 0)]
[(((x + 1) - 1 == 0) && (x == 0)) || (!((x + 1) - 1 == 0) && (x + 1 == x + 1))]
z = x + 1
[((z - 1 == 0) && (x == 0)) || (!(z - 1 == 0) && (z == x + 1))]
if (z - 1 == 0) then {
    [x == 0] [1 == x + 1]  y = 1 [y == x + 1]
} else {
    [z == x + 1]  y = z [y == x + 1]
}
[y == x + 1]
```

9.4 Toistolauseiden invariantit

- Ohjelmointikielten keskeisiä primitiivejä ovat toistolauseet, joiden avulla komentoja voidaan toistaa haluttu määrä.

  $$\mathtt{z=0\,;\,v=0\,;\,while(!(z==x))\,\{z=z+1\,;\,v=v+y\}}$$

- Ongelma: kuinka voitaisiin osoittaa toistorakenteita sisältävien algoritmien toimivuus kaikissa tilanteissa?
- Toistorakenteelle halutaan tyypillisesti todistaa *invariantti* eli ominaisuus, joka säilyy voimassa toistorakenteen suorituksen ajan.

Määritelmä. Toistolauseen $\mathtt{while}(B)\,\{C\}$ invariantti I on mikä tahansa Boolen lauseke siten, että $\models_p [B\ \&\&\ I]\ C\ [I]$.

Huomio. Invariantti I ei välttämättä ole jatkuvasti tosi komennon C suorituksen aikana, mutta ehdottomasti C:n suorituksen jälkeen.

Sopivan invariantin hakemisesta

- Osittaisen oikeellisuuden $\models_p [B_1]\ \mathtt{while}(B)\,\{C\}\ [B_2]$ todistaminen voi perustua sopivan invariantin I käyttöön:
  1. $\models_{\mathbb{Z}} B_1 \mathtt{->} I$,
  2. $\models_{\mathbb{Z}} (I\ \&\&\ !B) \mathtt{->} B_2$, ja
  3. $\models_p [I]\ \mathtt{while}(B)\,\{C\}\ [I\ \&\&\ !B]$.
- Kuten aiemminkin, oikeellisuustodistusta voi hakea takaperin:

A1. Valitaan I siten, että $\models_{\mathbb{Z}} (I\ \&\&\ !B) \mathtt{->} B_2$.

A2. Haetaan heikoin esiehto I' siten, että $\models_p [I']\ C\ [I]$.

A3. Osoitetaan $\models_{\mathbb{Z}} I\ \&\&\ B \mathtt{->} I'$, minkä nojalla $\models_p [I\ \&\&\ B]\ C\ [I]$ ja edelleen $\models_p [I]\ \mathtt{while}(B)\,\{C\}\ [I\ \&\&\ !B]$.

A4. Osoitetaan $\models_{\mathbb{Z}} B_1 \mathtt{->} I$.

Esimerkki. Osoitetaan edellä annetun kertolaskuohjelman `Multi` osittainen oikeellisuus eli $\models_p [\mathtt{true}]\ \mathtt{Multi}\ [\mathtt{v==x*y}]$.

$[\mathtt{true}]\ [\mathtt{0==0*y}]\ \mathtt{z=0}\ [\mathtt{0==z*y}]\ \mathtt{v=0}\ [\mathtt{v==z*y}]$ (A4)

$\mathtt{while(!(x==z))\,\{}$

$\quad[\mathtt{(v==z*y)\ \&\&\ !(x==z)}]$ (A3)

$\quad[\mathtt{v+y==(z+1)*y}]\ \mathtt{z=z+1}\ [\mathtt{v+y==z*y}]\ \mathtt{v=v+y}\ [\mathtt{v==z*y}]$ (A2)

$\mathtt{\}}$

$[\mathtt{(v==z*y)\ \&\&\ (x==z)}]\ [\mathtt{v==x*y}]$ (A1)

Huomio. Todistuksessa käytetty invariantti on $\mathtt{v==z*y}$.

Edellä esitetyt neljä todistusaskelta (A1)...(A4) on merkitty ylös.

Ohjelman suoritus päättyy, jos ja vain jos $\mathtt{!(x} < \mathtt{0)}$ on tosi.

9.5 Täysi oikeellisuus

- Tieto osittaisesta oikeellisuudesta ($\models_p [B_1]\ C\ [B_2]$) on hyödyllinen ainoastaan, mikäli komennon C suoritus todella päättyy.
- Täyden oikeellisuuden ($\models_t$) osoittamiseksi joudutaan todistamaan erikseen, että komennossa C esiintyvien toistolauseiden suoritus päättyy lopulta.
- Tätä varten tarvitaan vahvennettua päättelysääntöä

  $$\frac{[B_1\ \&\&\ B_2\ \&\&\ (E\mathtt{==n})]\ C\ [B_1\ \&\&\ (E\mathtt{<n})]}{[B_1]\ \mathtt{while}(B_2)\,\{C\}\ [B_1\ \&\&\ !B_2]},$$

  missä E on sopiva kokonaislukulauseke, $\mathtt{n}$ on (uusi) kokonaislukumuuttuja ja B_1 on vahvennettu invariantti $B\ \&\&\ (\mathtt{0<=}E)$.
- Näin lausekkeen E arvo pienenee jatkuvasti toistettaessa C:tä.
  $\Longrightarrow$ Suoritus päättyy vääjäämättä, koska $\mathtt{0<=}E$ säilyy voimassa.

Esimerkki. Osoitetaan $\models_t$ [0 <= x] `Multi` [v == x * y] seuraavasti:

```
[0 <= x] [(0 == 0 * y) && (0 <= x - 0)] z = 0 [(0 == z * y) && (0 <= x - z)]
v = 0 [(v == z * y) && (0 <= x - z)]
while(!(x == z)) {
    [(v == z * y) && (0 <= x - z) && (x - z == n) && !(x == z)]
    [(v + y == (z + 1) * y) && (0 <= x - (z + 1)) && (x - (z + 1) < n)]
    z = z + 1 [(v + y == z * y) && (0 <= x - z) && (x - z < n)]
    v = v + y [(v == z * y) && (0 <= x - z) && (x - z < n)]
}
[(v == z * y) && (0 <= x - z) && (x == z)] [v == x * y]
```

Huomioita. Lauseke, jonka arvo vähenee aidosti, on $x - z$.

Esiehto 0 <= x on yllä välttämätön, koska $\not\models_t$ [`true`] `Multi` [v == x * y]!