



PREDIKAATTOLOGIikka

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

1.1 Motivaatio

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

$C_d = "c \text{ on isompi kuin } d"$, $D_e = "d \text{ on isompi kuin } e"$, ...

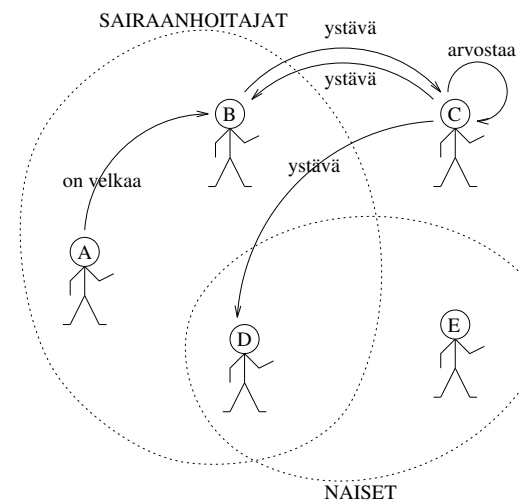
$(C_d \wedge D_e \rightarrow C_e) \wedge (C_e \wedge E_d \rightarrow C_d) \wedge (D_e \wedge E_c \rightarrow D_c) \wedge \dots$



1 Predikaattilogiikan kieli

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

Esimerkki. Alla on kuvattu joitain henkilöiden välisiä suhteita.



1.2 Predikaattilogiikan aakkosto

Predikaattilogiikan kielessä \mathcal{L} käytetään seuraavia symboleja:

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

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, \dots, t_n ovat termejä, niin myös $f(t_1, \dots, t_n)$ 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.

- 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 määrittellä 0-paikkaisten funktiosymbolien joukkona \mathcal{F}_0 .
- Joukon \mathcal{P}_0 symbolit vastaavat lauselogiikan atomisia lauseita.

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, \dots, t_n ovat termejä, niin $P(t_1, \dots, t_n)$ on *atomikaava*.
4. Muita atomikaavoja ei ole.

Esimerkki. Atomikaavoja ovat mm.

$$P(x_1), \quad Q, \quad x_1 = x_2, \quad g(x_1, x_2) = f(f(c_1)),$$

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

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



Määritelmä.

- Jokainen atomikaava ϕ on *kaava*.
- 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*.
- Muita kaavoja ei ole.

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

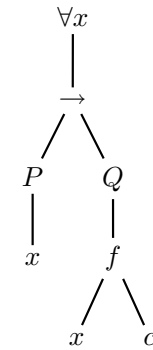
Symbolijoukkoihin $\mathcal{V}, \mathcal{C}, \mathcal{F}$ ja \mathcal{P} perustuva predikaattilogiikan *kieli* \mathcal{L} määritellään edellä annetuilla periaatteilla muodostettavissa olevien kaavojen joukkona.

Kaavojen jäsenyspuut

Predikaattilogiikan kaavoilla on yksikäsitteinen jäsenyspuu.

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

Kyseinen kaava on muodoltaan *universaalisti kvantifioitu kaava*.



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



Sopimukset sulkeiden käytöstä

- Konnektiivien presedenssiluokat ovat seuraavat:
 - \neg , $\forall v$ ja $\exists v$ (missä $v \in \mathcal{V}$) ovat vahvimmat konnektiivit.
 - \vee ja \wedge ovat näitä heikompia, mutta vahvempia kuin \rightarrow ja \leftrightarrow .
 - \rightarrow ja \leftrightarrow ovat heikoimmat konnektiivit.
- Lauselogiikan yhteydessä käyttöön otettuja 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

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

1.4 Kaavojen muodostaminen

- 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.
- Tutkitaan millaisia relaatioita objektien välillä on ja otetaan käyttöön näitä vastaavat predikaattisymbolit.
- Kuvataan relaatioiden väliset riippuvuudet kirjoittamalla niille määritelmät predikaattilogiikan kaavoina.

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

**Esimerkki.**

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

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

$\exists x P(x, h) =$

"on olemassa x , joka on hatun päällä",
eli "hatun päällä on jotakin".

$\exists x \forall y \neg P(y, x) =$

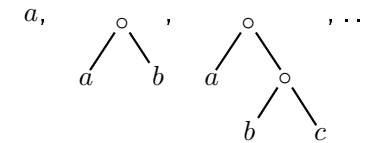
"on olemassa x siten, että mikään y ei ole x :n päällä",
eli "jonkin päällä ei ole mitään".

$\forall x (P(x, h) \rightarrow P(x, t)) =$

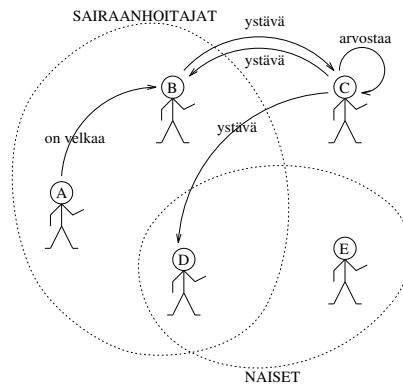
"kaikille x : jos x on hatun päällä, niin x on tuolin päällä",
eli "kaikki hatun päällä olevat ovat tuolin päällä".

Esimerkki. Funktiosymbolit tarjoavat tavan esittää induktiivisia tietorakenteita termien avulla.

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



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. 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):

$$\forall x \forall y \forall z \forall v (P(x, y) \wedge P(z, v) \rightarrow P(t(x, z), t(y, v))).$$

\Rightarrow 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 ($\neg, \rightarrow, \leftrightarrow, \vee, \wedge$) käsitellään vastaavalla tavalla (vrt. alilauseiden määritelmä lauselogiikan tapauksessa).

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

Määritelmä. Muuttujan x esiintymä kaavassa ϕ on *sidottu*, jos se sijaitsee kvantorin $\forall x$ (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:

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

Esimerkki. Kaavan $\exists x(P(x) \wedge \exists xQ(x))$ muuttujaesiintymät ovat sidotut.

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

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



Vapaat ja sidotut muuttujat kaavassa

Määritelmä. Olkoot $\exists x\psi$ ja $\forall x\phi$ predikaattilogiikan kaavoja. Aikaava ψ on kvantorin $\exists x$ *vaikutusalue* kaavassa $\exists x\psi$. Vastaavasti aikaava ϕ on kvantorin $\forall x$ vaikutusalue kaavassa $\forall x\phi$.

Esimerkki.

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

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

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 $\phi(y) = \exists x(P(x, y) \vee Q(y, x))$.

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

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

2. Olkoon $\psi(x) = \exists xP(x) \wedge Q(x)$.

- Sijoittamalla vakio c saadaan $\psi(c) = \exists xP(x) \wedge Q(c)$.



2 Predikaattilogiikan semantiikka

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

Huomioita.

- Joukon $U^n = \overbrace{U \times \dots \times U}^{n \text{ kpl}}$ alkioit ovat monikkoja (tai jonoja) $\langle a_1, \dots, a_n \rangle$, missä alkioit $a_1 \in U, \dots, a_n \in U$.
- 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).



2.1 Struktuurit

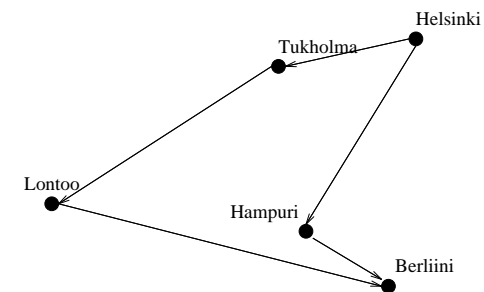
Predikaattilogiikassa totuusjakelet korvataan struktuureilla.

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 $f^{\mathcal{S}}: U^n \rightarrow U$.
 4. Predikaattisymbolin $P \in \mathcal{P}_n$ tulkintana on relaatio $P^{\mathcal{S}} \subseteq U^n$.

Struktuuri voidaan edelleen ymmärtää yhden asiaintilan kuvauksena.

Esimerkki.



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



2.2 Predikaattilogiikan totuusmääritelmä

Termien tulkinta struktuurissa

Määritelmä. Olkoon \mathcal{S} strukturi kielelle \mathcal{L} ja U strukturin \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, \dots, t_n nimeävät universumin U alkioita $t_1^{\mathcal{S}}, \dots, t_n^{\mathcal{S}}$ ja $f \in \mathcal{F}_n$, niin termi $f(t_1, \dots, t_n)$ *nimeää* universumin U alkion $f^{\mathcal{S}}(t_1^{\mathcal{S}}, \dots, t_n^{\mathcal{S}})$.

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

Kaavojen totuusarvojen laskeminen struktuurissa

Olkoon \mathcal{S} strukturi kielelle \mathcal{L} ja U strukturin \mathcal{S} universumi.

Määritelmä. Seuraavassa määritellään, milloin kaava $\phi \in \mathcal{L}$ on *tos* 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. $\mathcal{S} \models P(t_1, \dots, t_n) \iff \langle t_1, \dots, t_n \rangle^{\mathcal{S}}$ (eli jono $\langle t_1^{\mathcal{S}}, \dots, t_n^{\mathcal{S}} \rangle$) kuuluu tulkintaan $P^{\mathcal{S}}$ (yllä $n > 0$, $P \in \mathcal{P}_n$ ja t_1, \dots, 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. $\mathcal{S} \models \neg \alpha \iff \mathcal{S} \not\models \alpha$.



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

$$c^{\mathcal{S}} = 0,$$

$$f^{\mathcal{S}} = \text{seuraajafunktio, eli } f^{\mathcal{S}}(n) = n + 1, \text{ ja}$$

$$g^{\mathcal{S}} = \text{summafunktio, eli } g^{\mathcal{S}}(n, m) = n + m.$$

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

$$5. \mathcal{S} \models \alpha \wedge \beta \iff \mathcal{S} \models \alpha \text{ ja } \mathcal{S} \models \beta.$$

$$6. \mathcal{S} \models \alpha \vee \beta \iff \mathcal{S} \models \alpha \text{ tai } \mathcal{S} \models \beta.$$

$$7. \mathcal{S} \models \alpha \rightarrow \beta \iff \mathcal{S} \not\models \alpha \text{ tai } \mathcal{S} \models \beta.$$

$$8. \mathcal{S} \models \alpha \leftrightarrow \beta \iff \text{joko } \mathcal{S} \models \alpha \text{ ja } \mathcal{S} \models \beta, \text{ tai } \mathcal{S} \not\models \alpha \text{ ja } \mathcal{S} \not\models \beta.$$

$$9. \mathcal{S} \models \exists x \alpha(x) \iff$$

$$\mathcal{S}[x \mapsto a] \models \alpha(x) \text{ jollekin universumin } U \text{ alkioille } a \in U.$$

$$10. \mathcal{S} \models \forall x \alpha(x) \iff$$

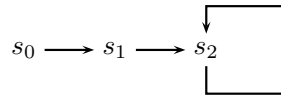
$$\mathcal{S}[x \mapsto a] \models \alpha(x) \text{ kaikille universumin } U \text{ alkioille } a \in U.$$

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



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. Tarkastellaan seuraavia opiskelijatietokannan relaatiotauluja:

Op (opiskelija)		Ko (koulutusohjelma)	
3310D	Taina	4358E	Tieto
4820H	Otso	3310D	Sähkö
9055Z	Elsi	9055Z	Kemia
4358E	Uljas	4820H	Rakennus
...

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



Esimerkki. (jatkoa)

4. Koska $\mathcal{S}[x \mapsto s_i] \models \exists y K(x, y)$ jokaiselle universumin alkion $s_i \in U$, 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!

2.3 Semanttiset peruskäsitteet

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

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 $\mathcal{S} \models \exists x \forall y P(x, y).$

Esimerkki. $\models \forall x P(x) \vee \neg \forall x P(x)$

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 $\mathcal{S} \models \forall x P(x) \vee \neg \forall x P(x).$

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

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



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:

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

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

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.

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

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

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



Esimerkki. Luokitellaan viikonpäiviä seuraavalla lausejoukolla:

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

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

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

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

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

3 Normaalimuodot

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



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.
- $\models \alpha \iff \emptyset \models \alpha$.
- $\{\alpha_1, \dots, \alpha_n\} \models \alpha \iff \models \alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \alpha$.

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

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

3.1 Prenex-normaalimuoto

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

$$Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \phi,$$

missä jokainen Q_i on jompikumpi kvanttoreista (\forall tai \exists) ja alikaava ϕ ei sisällä kvanttoreita.

Esimerkki. Seuraavat lauseet ovat prenex-normaalimuodossa:

$$P(a), \forall x P(x), \forall x \exists y P(x, y) \text{ ja } \forall x \exists y \forall z \forall w (P(x, y, z) \rightarrow (Q(y, z, w) \rightarrow R(z, w, x))).$$

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



Lauseiden muuttaminen prenex-normaaliinmuotoon

Mikä tahansa predikaattilogiikan lause voidaan muuttaa prenex-normaaliinmuotoon 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:

$$\neg\neg\varphi \rightsquigarrow \varphi$$

$$\neg(\varphi \wedge \psi) \rightsquigarrow \neg\varphi \vee \neg\psi$$

$$\neg\forall y\varphi \rightsquigarrow \exists y\neg\varphi$$

$$\neg(\varphi \vee \psi) \rightsquigarrow \neg\varphi \wedge \neg\psi$$

$$\neg\exists y\varphi \rightsquigarrow \forall y\neg\varphi$$

Esimerkki. Muunnetaan seuraava lause prenex-normaaliinmuotoon:

$$\forall x(P(x) \rightarrow \exists zR(z, x)) \rightarrow \exists xQ(x)$$

$$\rightsquigarrow \neg\forall x(P(x) \rightarrow \exists zR(z, x)) \vee \exists xQ(x)$$

$$\rightsquigarrow \neg\forall x(\neg P(x) \vee \exists zR(z, x)) \vee \exists xQ(x)$$

$$\rightsquigarrow \exists x\neg(\neg P(x) \vee \exists zR(z, x)) \vee \exists xQ(x)$$

$$\rightsquigarrow \exists x(\neg\neg P(x) \wedge \neg\exists zR(z, x)) \vee \exists xQ(x)$$

$$\rightsquigarrow \exists x(P(x) \wedge \forall z\neg R(z, x)) \vee \exists xQ(x)$$

$$\rightsquigarrow \exists x(\exists x(P(x) \wedge \forall z\neg R(z, x)) \vee Q(x))$$

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

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

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



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!)

3.2 Konjunkttiivinen 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, \dots, 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 konjunkttiivisessa normaalimuodossa, mikäli se on on prenex-normaaliinmuodossa $Q_1x_1Q_2x_2\cdots Q_nx_n\phi$, missä kvanttoireita sisältämätön osa $\phi = \phi_1 \wedge \cdots \wedge \phi_n$ ja jokainen konjunktion jäsen ϕ_i on muodoltaan literaalien disjunktio.

Esimerkki. Seuraava lause on konjunkttiivisessa 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-normaali muodon kvanttoireita sisältämätön osa voidaan järjestää konjunkttiiviseen normaalimuotoon seuraavasti:

$$\varphi \vee (\phi \wedge \psi) \rightsquigarrow (\varphi \vee \phi) \wedge (\varphi \vee \psi)$$

$$(\phi \wedge \psi) \vee \varphi \rightsquigarrow (\phi \vee \varphi) \wedge (\psi \vee \varphi)$$

- Näin ollen voimme todeta seuraavan tuloksen:

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

Esimerkki. Muunnetaan edellä johdettu prenex-normaali muoto edelleen konjunkttiiviseksi normaalimuodoksi:

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

Eksistenssi kvanttorien eliminointi yleisessä tapauksessa

Olkoon $\vec{Q}x\phi$ prenex-normaali muodossa ja $\exists x$ kvanttorisekvenssin $\vec{Q}x$ ensimmäinen eksistenssi kvanttori.

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

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



3.3 Eksistenssi kvanttorien eliminointi

Esimerkki. Tarkastellaan kahta kokonaislukuja koskevaa väittämää:

- Summafunktiolla on vasen identiteetti:

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

Identiteettialkio voidaan nimetä vakiosymbolilla 0:

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

- Jokaisella kokonaisluvulla on vastaluku:

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

Vastalukufunktio voidaan nimetä funktiosymbolilla $-$:

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

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

Esimerkki. Suoritetaan Skolemointi seuraaville lauseille:

$$\exists x P(x) \rightsquigarrow P(c)$$

$$\exists x \forall y \exists x P(x, y) \rightsquigarrow \forall y P(f(y), y)$$

$$\forall x \exists y (P(x, y) \wedge Q(y, x)) \rightsquigarrow \forall x (P(x, f(x)) \wedge Q(f(x), x))$$

$$\forall x \exists y \forall z \exists w P(w, z, y, x) \rightsquigarrow \forall x \forall z P(g(x, z), z, f(x), x)$$

$$\exists x \exists y \forall z P(x, y, z, x) \rightsquigarrow \forall z P(c_1, c_2, z, c_1)$$

$$\forall x \exists y \exists z P(x, y, z) \rightsquigarrow \forall x P(x, f_1(x), f_2(x))$$



Skolemoinnin loogiset ominaisuudet

Väite. Prenex-normaali muodossa oleva lause φ on toteutuva \iff lauseen φ skolemoitu muoto φ' on toteutuva.

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

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

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

Vastamalli \mathcal{S} : universumi $U = \{1, 2\}$, $c^{\mathcal{S}} = 1$ ja $P^{\mathcal{S}} = \{2\}$.

Nyt $\mathcal{S} \models \exists xP(x)$, mutta $\mathcal{S} \not\models P(c)$.

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

Huomio. Jos lause on muodoltaan *konjunktio* $\phi_1 \wedge \dots \wedge \phi_n$, on mahdollista saattaa konjunktin jäsenet ϕ_1, \dots, ϕ_n klausuulimuotoon *erikseen*. Muista myös, että $\neg(\phi_1 \wedge \dots \wedge \phi_n \rightarrow \phi) \equiv \phi_1 \wedge \dots \wedge \phi_n \wedge \neg\phi$.

Esimerkki. $\forall xP(x) \wedge \exists xQ(x)$. Konjunktin 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 universaalikvanttoita (mikäli mahdollista).

Esimerkki. Siis $\forall xP(x) \vee \exists xQ(x)$ 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 \implies klausuulimuodosta tulee rakenteeltaan yksinkertaisempi.



3.4 Lauseiden klausuulimuoto predikaattilogiikassa

Mille tahansa lauseelle voidaan hakea klausuulimuoto seuraavasti:

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

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

$$\rightsquigarrow \forall x\exists z((P(x) \wedge \neg Q(x, z)) \vee R(x)) \quad (1)$$

$$\rightsquigarrow \forall x\exists z((P(x) \vee R(x)) \wedge (\neg Q(x, z) \vee R(x))) \quad (2)$$

$$\rightsquigarrow \forall x((P(x) \vee R(x)) \wedge (\neg Q(x, f(x)) \vee R(x))) \quad (3)$$

$$\rightsquigarrow \{\{P(x), R(x)\}, \{\neg Q(x, f(x)), R(x)\}\}. \quad (4)$$

4 Semanttiset taulut predikaattilogiikalle

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



4.1 Taulusäännöt kvantorien käsittelyyn

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

$T\exists x\varphi(x)$	$E\forall x\varphi(x)$
$T\varphi(c)$	$E\varphi(c)$
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).

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

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



- 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).

$T\forall x\varphi(x)$	$E\exists x\varphi(x)$
$T\varphi(t)$	$E\varphi(t)$
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.

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 .

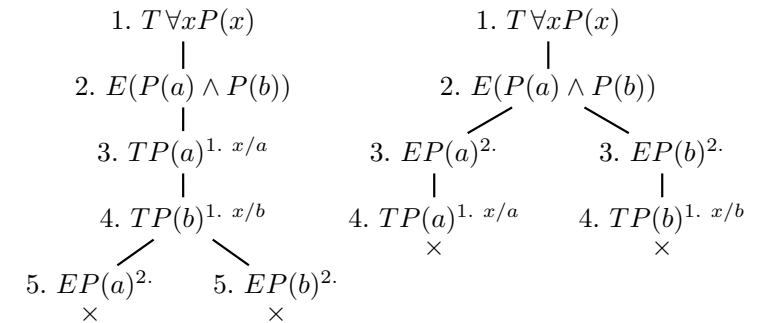


4.3 Ohjeita taulutodistusten laadintaan

- Lauseen rakenne määrää edelleen, mitä taulusääntöä tulee käyttää (jäsennykspuun juuressa oleva konnektiivi).
- Solmujen hajoittamisjärjestyksellä voi usein vaikuttaa taulun kokoon (haarautumista kannattaa välttää).
- Jälkimmäisissä kvantorisäännöissä 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.

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

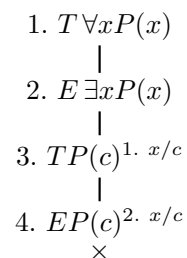
Esimerkki. $\{\forall xP(x)\} \vdash P(a) \wedge P(b)$



Taulutodistusten erityispiirteitä predikaattilogiikan tapauksessa

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

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



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

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 kvantorisäännöillä 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:

$$\begin{array}{l}
 1. T \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \rightarrow P(x, z)) \\
 \quad | \\
 2. TP(a, b) \\
 \quad | \\
 3. TP(b, c) \\
 \quad | \\
 4. EP(a, c) \\
 5. T \forall y \forall z (P(a, y) \wedge P(y, z) \rightarrow P(a, z))^{1. x/a} \\
 \quad | \\
 6. T \forall z (P(a, b) \wedge P(b, z) \rightarrow P(a, z))^{5. y/b} \\
 \quad | \\
 7. T(P(a, b) \wedge P(b, c) \rightarrow P(a, c))^{6. z/c} \\
 \quad / \quad \backslash \\
 8. E(P(a, b) \wedge P(b, c))^{7.} \quad 8. TP(a, c)^{7.} \\
 \quad \backslash \quad / \\
 9. EP(a, b)^{8.} \quad 9. EP(b, c)^{8.} \\
 \quad \times \quad \quad \quad \times
 \end{array}$$

© 2007 TKK / Tietojenkäsittelyteorian laboratorio

Kvanttorisekvensien käsittely

Jatkossa sallimme seuraavien johdettujen taulusääntöjen käytön kvanttorisekvensien käsittelyssä:

$T \forall x_1 \dots \forall x_n \varphi(x_1, \dots, x_n)$	$E \exists x_1 \dots \exists x_n \varphi(x_1, \dots, x_n)$
$T \varphi(t_1, \dots, t_n)$	$E \varphi(t_1, \dots, t_n)$

$T \exists x_1 \dots \exists x_n \varphi(x_1, \dots, x_n)$	$E \forall x_1 \dots \forall x_n \varphi(x_1, \dots, x_n)$
$T \varphi(c_1, \dots, c_n)$	$E \varphi(c_1, \dots, c_n)$

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

© 2007 TKK / Tietojenkäsittelyteorian laboratorio



4. Solmujen keskinäinen hajoitusjärjestys voi olla ratkaisevassa roolissa.

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

$$\begin{array}{l}
 1. T \forall x P(x) \\
 \quad | \\
 2. T \forall x (P(x) \rightarrow Q(x)) \\
 \quad | \\
 3. T Q(a) \\
 \quad | \\
 4. E \forall x Q(x) \\
 \quad | \\
 5. EQ(c)^{4. x/c} \\
 \quad | \\
 6. TP(c)^{1. x/c} \\
 \quad | \\
 7. T(P(c) \rightarrow Q(c))^{2. x/c} \\
 \quad / \quad \backslash \\
 8. EP(c)^{7.} \quad 8. TQ(c)^{7.} \\
 \quad \times \quad \quad \quad \times
 \end{array}$$

© 2007 TKK / Tietojenkäsittelyteorian laboratorio

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. hylkävään tilaan e (ei), jos syöte φ ei ole pätevä.

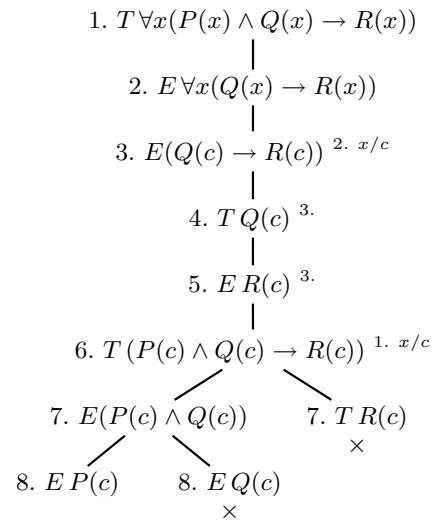
Huomioita.

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

© 2007 TKK / Tietojenkäsittelyteorian laboratorio

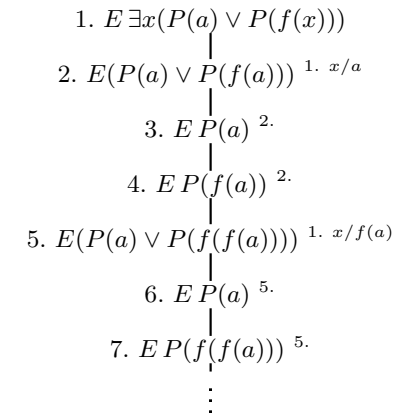


Esimerkki. $\{\forall x(P(x) \wedge Q(x) \rightarrow R(x))\} \not\models \forall x(Q(x) \rightarrow R(x))$.



© 2007 TKK / Tietojenkäsittelyteorian laboratorio

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



Edellytyksenä on muuttujattomien termien tulkinta samalle U :n alkioille.

© 2007 TKK / Tietojenkäsittelyteorian laboratorio



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

© 2007 TKK / Tietojenkäsittelyteorian laboratorio

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

© 2007 TKK / Tietojenkäsittelyteorian laboratorio



5 Tietämyksen esittämisestä

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

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

$$\{ \text{linkki}(a, b), \text{linkki}(b, c), \text{linkki}(d, e), \\ \forall x \text{ yhteys}(x, x), \\ \forall x \forall y (\text{linkki}(x, y) \rightarrow \text{yhteys}(x, y)), \\ \forall x \forall y (\text{yhteys}(x, y) \rightarrow \text{yhteys}(y, x)), \\ \forall x \forall y \forall z (\text{yhteys}(x, y) \wedge \text{yhteys}(y, z) \rightarrow \text{yhteys}(x, z)) \}.$$

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



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 $\text{Cn}(\Sigma) = \{ \phi \in \mathcal{L} \mid \phi \text{ on lause ja } \Sigma \models \phi \}$. Nyt

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

Tietämyksen esittämisen problematiikkaa (kertaus)

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

\Rightarrow 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. $\Sigma \not\models \text{yhteys}(a, e)$.

- 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 $\Sigma' = \Sigma \cup \{\text{linkki}(d, c)\}$, jolle $\Sigma' \models \text{yhteys}(a, e)$.
- Huomaa, että $\mathcal{S} \not\models \text{linkki}(d, c)$, joten vastamallimme rajautuu pois.

- Predikaattilogiikalla annetut määritelmät ovat korkeintaan yhtä monimutkaisia kuin edellä kuvattu normaalimuoto.
- Jos jokainen kvanttori 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, \dots, Q_k ja P argumentteina on termijonot $\vec{t}_1, \dots, \vec{t}_k$ ja \vec{t} , jotka rakentuvat siis vakiosymboleista, muuttujasymboleista x_1, \dots, x_n ja funktiosymboleista.

- Määritelmiä voidaan usein kirjoittaa tähän muotoon: mietitään millä ehdoilla $Q_1(\vec{t}_1), \dots, Q_k(\vec{t}_k)$ voidaan päätellä predikaattia P koskeva väittämä $P(\vec{t})$. Ko. muotoa olevia lauseita saatetaan tarvita useita.



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

$$Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \psi$$

missä kukin kvanttori 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

$$\begin{aligned} & \neg Q_1(\vec{t}_1) \vee \cdots \vee \neg Q_k(\vec{t}_k) \vee P_1(\vec{s}_1) \vee \cdots \vee P_l(\vec{s}_l) \\ \equiv & Q_1(\vec{t}_1) \wedge \cdots \wedge Q_k(\vec{t}_k) \rightarrow P_1(\vec{s}_1) \vee \cdots \vee P_l(\vec{s}_l). \end{aligned}$$

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

$$\text{tartuntavaarassa}(x) = \text{"henkilö } x \text{ on tartuntavaarassa"}.$$

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

$$\forall x \forall y \dots (Q_1(\vec{t}_1) \wedge \dots \wedge Q_k(\vec{t}_k) \rightarrow \text{tartuntavaarassa}(x)).$$



- Näin saadaan muodostettua lauseet

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

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

- Kysymyksessä on itse asiassa tartuntavaarassa-predikaatin induktiivinen (rekursiivinen) määritelmä. Lauseista ensimmäinen vastaa perustapausta ja jälkimmäinen induktioaskelta.

- Lisäksi voidaan todeta tapaamiset symmetrisiksi:

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

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

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 $t_1 \in H, \dots, t_n \in H$ pätee seuraavaa: $\langle t_1, \dots, t_n \rangle \in P^* \iff \Sigma_P \models P(t_1, \dots, t_n)$.
- Tällainen *positivistinen* määritelmä ei ole välttämättä *täydellinen* eli määritelmä ei takaa, että $\Sigma_P \models \neg P(t_1, \dots, t_n)$ mikäli $\langle t_1, \dots, t_n \rangle \notin P^*$.



Määritelmien käyttö konkreettisessa päätelyssä

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

Näin saadaan lausejoukko

$$\Sigma = \{ \forall x \forall y (\text{tapaa}(x, y) \wedge \text{sairastaa}(y) \rightarrow \text{tartuntavaarassa}(x)),$$

$$\forall x \forall y (\text{tapaa}(x, y) \wedge \text{tartuntavaarassa}(y) \rightarrow \text{tartuntavaarassa}(x)),$$

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

$$\text{tapaa}(\text{Lyyli}, \text{Hemmo}), \text{tapaa}(\text{Lyyli}, \text{Erkki}), \text{sairastaa}(\text{Erkki}) \}.$$

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

Esimerkki. Palataan taas radioverkkoesimerkin lausejoukkoon $\Sigma =$

$$\{ \text{linkki}(a, b), \text{linkki}(b, c), \text{linkki}(d, e),$$

$$\forall x \text{ yhteys}(x, x),$$

$$\forall x \forall y (\text{linkki}(x, y) \rightarrow \text{yhteys}(x, y)),$$

$$\forall x \forall y (\text{yhteys}(x, y) \rightarrow \text{yhteys}(y, x)),$$

$$\forall x \forall y \forall z (\text{yhteys}(x, y) \wedge \text{yhteys}(y, z) \rightarrow \text{yhteys}(x, z)) \}.$$

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



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:

$$\forall x(T(x) \rightarrow \phi(x)) \text{ ja } \exists x(T(x) \wedge \phi(x)).$$

- Tyypeillä voi olla erilaisia suhteita:

- Erillisuus: $\forall x \neg(\text{henkilö}(x) \wedge \text{tauti}(x))$.
- 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:

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

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



Esimerkki. Lisätään tartuntavaaraesimerkkiin tyyppipredikaatteja:

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

$$\forall x \forall y \forall z(\text{henkilö}(x) \wedge \text{henkilö}(y) \wedge \text{tapaa}(x, y) \wedge$$

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

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

$$\text{tartuntavaarassa}(y, z) \rightarrow \text{tartuntavaarassa}(x, z)).$$

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:

$$\forall x(\text{solmu}(x) \rightarrow \exists y(\text{väri}(y) \wedge \text{väritetty}(x, y)))$$

$$\equiv \forall x \exists y(\text{solmu}(x) \rightarrow \text{väri}(y) \wedge \text{väritetty}(x, y)).$$

- 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

- Rajoitetaan jatkossa predikaattilogiikan kielisiin \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
 - jokin universumin U alkio $a \in U$ on useamman vakion c_1, \dots, c_n ($n > 1$) nimeämä: $c_1^{\mathcal{S}} = \dots = c_n^{\mathcal{S}} = a$.
 - 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.

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, \dots .

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

\Rightarrow Universumissa oltava vähintään 3 henkilöä.



Nimien yksikäsitteisyys

- Vastaava käsite englanniksi on *unique names assumption* (UNA).
- Kun kielessä on äärellinen määrä vakiosymboleita c_1, \dots, c_n , riittää lisätä muotoa

$$\neg(c_i = c_j)$$
 olevat lauseet, missä $i \in \{1, \dots, n\}$, $j \in \{1, \dots, n\}$ 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:

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

Nimien kattavuus

- Vastaava käsite englanniksi on *domain closure assumption* (DCA).
- Kun kielessä on äärellinen määrä vakiosymboleita c_1, \dots, c_n , riittää lisätä seuraavaa muotoa oleva lause:

$$\forall x(x = c_1 \vee \dots \vee x = c_n).$$

- 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

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

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

U_i	Lyyli ^{\mathcal{S}_i}	Hemmo ^{\mathcal{S}_i}	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

\Rightarrow Universumissa voi olla korkeintaan 3 henkilöä.

Yhtäsuuruuspredikaatin määritelmä

Jos yhtäsuuruuspredikaattia sisältäviä lauseita käytetään määritelmässä, 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. Transitivisyys: $\forall x \forall y \forall z((x = y) \wedge (y = z) \rightarrow (x = z))$.
4. Sijoitettavuus (kaikille predikaateille $P \in \mathcal{P}_n$):

$$\forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n$$

$$(P(x_1, \dots, x_n) \wedge (x_1 = y_1) \wedge \dots \wedge (x_n = y_n) \rightarrow P(y_1, \dots, y_n)).$$

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



Esimerkki. Tarkastellaan vielä edeltävien lausejoukkojen unionin

$$\Sigma_{UNA} \cup \Sigma_{DCA} = \{ \neg(\text{Lyyli} = \text{Hemmo}), \neg(\text{Lyyli} = \text{Erkki}), \\ \neg(\text{Hemmo} = \text{Erkki}), \\ \forall x(x = \text{Lyyli} \vee x = \text{Hemmo} \vee x = \text{Erkki}) \}$$

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

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

\Rightarrow Universumissa on oltava täsmälleen 3 henkilöä.

5.4 Negatiiviset ehdot ja johtopäätökset

- Tarkastellaan muotoa

$$\forall x_1 \forall x_2 \dots \forall x_n (Q_1(\vec{t}_1) \wedge \dots \wedge Q_k(\vec{t}_k) \rightarrow P(\vec{t}))$$

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.

$$\forall x(\neg \text{sairastaa}(x) \wedge \neg \text{tartuntavaarassa}(x) \rightarrow \text{turvassa}(x))$$

$$\equiv \forall x(\text{sairastaa}(x) \vee \text{tartuntavaarassa}(x) \vee \text{turvassa}(x)).$$

- 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

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

kaikille kielen \mathcal{L} muuttujattomille termeille t_1, \dots, 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 $\Sigma_P \models P(t_1, \dots, t_n)$ että $\Sigma_P \models \neg P(t_1, \dots, t_n)$ kaikille muuttujattomille termeille $t_1 \in H, \dots, t_n \in H$.
- Jos predikaatin $P \in \mathcal{P}_n$ määritelmä Σ_P on täydellinen ja $\Sigma_P \not\models P(t_1, \dots, t_n)$ joillekin muuttujattomille termeille $t_1 \in H, \dots, t_n \in H$, niin Σ_P ei ole ristiriitainen ja $\Sigma_P \models \neg P(t_1, \dots, t_n)$.

Predikaatin määritelmän täydentäminen

Esimerkki. Täydennetään edellisen esimerkin predikaattien määritelmät.

- Yhtäsuuruuspredikaatin osalta riittää todeta nimien yksikäsitteisyys:

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

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

$$\forall x(\text{tapaa}(x, y) \leftrightarrow (x = \text{Lyyli} \wedge y = \text{Hemmo}) \vee \\ (x = \text{Hemmo} \wedge y = \text{Lyyli}) \vee \\ (x = \text{Lyyli} \wedge y = \text{Erkki}) \vee \\ (x = \text{Erkki} \wedge y = \text{Lyyli})).$$



Esimerkki. Tarkastellaan *muunnelmaa* tartuntavaara-esimerkistä:

$$\Sigma = \{ \forall x \forall y (\text{tapaa}(x, y) \wedge \text{sairastaa}(y) \rightarrow \text{tartuntavaarassa}(x)), \\ \forall x (\neg \text{sairastaa}(x) \wedge \neg \text{tartuntavaarassa}(x) \rightarrow \text{turvassa}(x)), \\ \text{tapaa}(\text{Lyyli}, \text{Hemmo}), \text{tapaa}(\text{Hemmo}, \text{Lyyli}), \\ \text{tapaa}(\text{Lyyli}, \text{Erkki}), \text{tapaa}(\text{Erkki}, \text{Lyyli}), \text{sairastaa}(\text{Erkki}) \}.$$

- 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 =).

- Predikaattien tartuntavaarassa ja turvassa määritelmät voidaan kirjoittaa vastaavasti ekvivalensseiksi:

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

- Täydennetyillä määritelmillä on haluamme loogiset seuraukset:

sairastaa	tartuntavaarassa	turvassa
$\neg \text{sairastaa}(\text{Lyyli})$	$\text{tartuntavaarassa}(\text{Lyyli})$	$\neg \text{turvassa}(\text{Lyyli})$
$\neg \text{sairastaa}(\text{Hemmo})$	$\neg \text{tartuntavaarassa}(\text{Hemmo})$	$\text{turvassa}(\text{Hemmo})$
$\text{sairastaa}(\text{Erkki})$	$\neg \text{tartuntavaarassa}(\text{Erkki})$	$\neg \text{turvassa}(\text{Erkki})$

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



Esimerkki. Tarkastellaan vastaavaa konstruktiota lausejoukolla

$$\Sigma = \{ \text{tuntee1}(\text{Lyyli}, \text{Lyyli}), \text{tuntee1}(\text{Hemmo}, \text{Hemmo}), \\ \forall x \forall y (\text{tuntee1}(x, y) \rightarrow \text{tuntee2}(x, y)), \\ \forall x \forall y (\text{tuntee2}(y, x) \rightarrow \text{tuntee2}(x, y)) \}.$$

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

$$\Sigma' = \{ \neg(\text{Lyyli} = \text{Hemmo}), \quad \forall x (x = \text{Lyyli} \vee x = \text{Hemmo}), \\ \forall x \forall y (\text{tuntee1}(x, y) \leftrightarrow (x = \text{Lyyli} \wedge y = \text{Lyyli}) \vee \\ (x = \text{Hemmo} \wedge y = \text{Hemmo})), \\ \forall x \forall y (\text{tuntee2}(x, y) \leftrightarrow \text{tuntee1}(x, y) \vee \text{tuntee2}(y, x)) \}.$$

6 Herbrandin teoreema

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



- Yllättäen täydennetyistä määritelmistä ei seuraa loogisesti $\neg \text{tuntee2}(\text{Lyyli}, \text{Hemmo})$ eikä $\neg \text{tuntee2}(\text{Hemmo}, \text{Lyyli})$.

- Lausejoukolla Σ' on seuraava epäintuitiivinen malli \mathcal{S} :

$$\text{Universumi } U = \{h_1, h_2\},$$

$$\text{Lyyli}^{\mathcal{S}} = h_1, \text{Hemmo}^{\mathcal{S}} = h_2,$$

$$\text{tuntee1}^{\mathcal{S}} = \{\langle h_1, h_1 \rangle, \langle h_2, h_2 \rangle\} \text{ ja}$$

$$\text{tuntee2}^{\mathcal{S}} = \{\langle h_1, h_1 \rangle, \langle h_1, h_2 \rangle, \langle h_2, h_1 \rangle, \langle h_2, h_2 \rangle\}.$$

- Kyseinen struktuuri \mathcal{S} on vastamalli, koska

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

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

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

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

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

Määritelmä. Kielen \mathcal{L} Herbrand-strukturi \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

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

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



6.2 Herbrand-strukturi ja -mallit

Määritelmä. Kielen \mathcal{L} *Herbrand-strukturi* on strukturi \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 $f^{\mathcal{H}} : H^n \rightarrow H$, joka kuvaa muuttujattomat termit $t_1 \in H, \dots, t_n \in H$ muuttujattomaksi termiksi $f(t_1, \dots, t_n) \in H$, ja
5. jokaisen predikaattisymbolin $P \in \mathcal{P}_n$ tulkintana on $P^{\mathcal{H}} \subseteq H^n$.

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

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-strukturien \mathcal{H} määrittelemisen Herbrand-kannan B osajoukkoina: jokaiselle $P \in \mathcal{P}_n$ pätee

$$P(t_1, \dots, t_n) \in \mathcal{H} \iff \langle t_1, \dots, t_n \rangle \in P^{\mathcal{H}}.$$

- Herbrand-strukturille \mathcal{H} voidaan antaa myös literaaliesitys:

$$\text{lit}(\mathcal{H}) = \{ P(a), Q(a), \neg R(a, a), \neg P(f(a)), Q(f(a)), R(a, f(a)), \neg R(f(a), a), \neg R(f(a), f(a)), \dots \}.$$



6.3 Herbrandin teoreema

- Rajoitutaan tarkastelemaan klausuulijoukkoja.
- Merkintä $C(x_1, \dots, x_n)$ tarkoittaa muuttujat x_1, \dots, x_n sisältävää klausuulia $\{P_1(\vec{t}_1), \dots, P_k(\vec{t}_k), \neg Q_1(\vec{s}_1), \dots, \neg Q_l(\vec{s}_l)\}$.
- Klausuuli $C(x_1, \dots, x_n)$ vastaa universaalisti kvantifioitua lausetta $\forall x_1 \dots \forall x_n \phi_C(x_1, \dots, x_n)$, missä $\phi_C(x_1, \dots, x_n)$ on klausuulin $C(x_1, \dots, x_n)$ esitys literaalien disjunktiona.
- Klausuulijoukolle S voidaan määritellä Herbrand-struktuurit samaan tapaan kuin lausejoukoillekin.
- Klausuulijoukko S voidaan *instantioida* vastaavan Herbrand-universumin H_S suhteen seuraavasti.

Väite. (Herbrandin 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 rajoittaa *syntaktisiin malleihin* (Herbrand-malleihin) mielivaltaisten mallien sijaan. Herbrandin teoreema johtaa myös naiiviin proseduriin 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).



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

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

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 *ilmaisvoimassa* (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 $S' = \{ \{P\}, \{ \neg P, Q \}, \{ \neg Q, Q \} \}$.

7.1 Substituutiot

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

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

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

1. korvattavat muuttujat x_1, \dots, 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*.



7 Unifikaatio

- Substituutiot
- Yleisimmät unifioidit
- Unifikaatioalgoritmi

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 $\theta = \{x_1/t_1, \dots, x_n/t_n\}$ 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 substituuio $\theta = \{x/y, y/x, z/x, v/f(z), w/g(f(y), c)\}$.

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

Määritelmä. Olkoot $\theta = \{x_1/t_1, \dots, x_n/t_n\}$ ja

$\lambda = \{y_1/u_1, \dots, y_m/u_m\}$ kaksi substituuotiota.

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

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

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 $\{x/f(b), z/y\}$.

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

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



7.2 Yleisimmät unifioijat

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

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)

7.3 Unifikaatioalgoritmi

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

Määritelmä. Olkoon S ei-tyhjä joukko johonkin predikaattisymboliin P perustuvia atomikaavoja $\{P(\vec{t}_1), \dots, P(\vec{t}_n)\}$.

1. Joukon S **erokohta** on vasemmalta oikealle siirryttäessä ensimmäinen kohta, jossa joukon S atomikaavojen merkkijonoesityksissä on jokin eroavaisuus.
2. Joukon S **eroujoukkoon** $D(S)$ kuuluvat atomikaavojen $P(\vec{t}_1), \dots, P(\vec{t}_n)$ erokohdasta alkavat termit u_1, \dots, u_n .

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



Unifikaatioalgoritmi ei-tyhjälle atomikaavojen joukolla 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.

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 $S\sigma = \{P(g(a), f(g(a)))\} = S_2$.



Väite. Olkoon S äärellinen ei-tyhjä joukko atomikaavoja.

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

Esimerkki. Lasketaan unifikaatioalgoritmillä 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 yksialkioinen, jatketaan.

8 Resoluutiosääntö ja -todistukset

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



8.1 Resoluutiosääntö predikaattilogiikan tapauksessa

Määritelmä. Olkoot

$$C_1 = C'_1 \sqcup \{P(\vec{t}_1), \dots, P(\vec{t}_n)\} \text{ ja } C_2 = C'_2 \sqcup \{\neg P(\vec{u}_1), \dots, \neg P(\vec{u}_m)\}$$

kaksi klausuulia,

1. joissa *ei esiinny yhteisiä* muuttujia ja
2. joissa esiintyvien atomikaavojen joukko

$$\{P(\vec{t}_1), \dots, P(\vec{t}_n), P(\vec{u}_1), \dots, P(\vec{u}_m)\}$$

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 $A \cup B$.

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, \dots, C_n tyhjälle klausuulille $C_n = \square$) $\iff S$ on toteutumaton.

Todistus. Sivuutetaan.



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.

Klausuulien $\{P(x_1), P(y_1)\}$ 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 \square (joukolle $\{P(x_1), P(y_1), P(x_2), P(y_2)\}$ MGU $\sigma = \{y_1/x_1, x_2/x_1, y_2/x_1\}$).

Esimerkki. Osoitetaan predikaattilogiikan lausejoukko

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

toteutumattomaksi. Haetaan lauseille ensin klausuuliesitykset:

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

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

$\implies S_1 \cup S_2$ on toteutumaton $\implies \Sigma$ 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: $\models \phi \iff \text{KM}(\{\neg\phi\})$:lle löytyy hylkäys.
2. Ekvivalenssi: $\phi \equiv \psi \iff \text{KM}(\{\neg(\phi \leftrightarrow \psi)\})$:lle löytyy hylkäys.
3. Looginen seuraavuus: $\Sigma \models \phi \iff$ klausuulijoukolle $\text{KM}(\Sigma \cup \{\neg\phi\})$ löytyy hylkäys.

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

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

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

loogiseksi seuraukseksi. Haetaan tarvittavat klausuulimuodot:

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

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



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

- Haetaan lauseen negaatiolle klausuulimuoto:

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

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

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. \square | 3,6, MGU ϵ |

- Yleisimpien unifioidien kompositio $\{x/y\}\{y/c\}\epsilon = \{x/c, y/c\}$.
- Täten S on toteutumaton
 - $\implies \Sigma \cup \{\neg \exists x(E(x) \wedge K(x))\}$ on toteutumaton
 - $\implies \exists x(E(x) \wedge K(x))$ 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 unifioidit (MGU:t) näkyviin.
- Ellet kirjoita todistusta binääripuun muotoon, numeroi klausuulit ja ilmoita, mistä klausuuleista mikin klausuuli on johdettu.
- Laske yleisimpien unifioidien kompositio selvittääksesi kyselyssä esiintyvillä muuttujilla arvot.

- Johdetaan lauseille klausuulimuodot:

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

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

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

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

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

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



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 alkiaina 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. $\forall y \forall v \forall z \forall x (L(y, v, z) \rightarrow L(c(x, y), v, c(x, z)))$.

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*).

Joitain PROLOGin erityispiirteitä

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

$$\{N(0)\} \rightsquigarrow n(0).$$

$$\{N(s(x)), \neg N(x)\} \rightsquigarrow n(s(X)) \text{ :- } n(X).$$

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

- Tyypillinen valintafunktio valitsee maaliklausulin 1. atomin.

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

8.4 Kytkenät logiikkaohjelmointiin (PROLOG)

Laskenta perustuu *järjestettyjen klausuulien* väliseen resoluutioon.

Määritelmä. Olkoon $G = \{\neg B_1(\vec{u}_1), \dots, \neg B_m(\vec{u}_m)\}$ kyselyn negaatiota vastaava järjestetty *maaliklausuuli* ja $C = \{A(\vec{t}), \neg A_1(\vec{t}_1), \dots, \neg A_n(\vec{t}_n)\}$ järjestetty *ohjelmaklausuuli* (ohjelman sääntö) siten, että

- klausuuleilla G ja C ei ole yhteisiä muuttujia ja
- 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

$$G' = \{ \neg B_1(\vec{u}_1), \dots, \neg B_{i-1}(\vec{u}_{i-1}), \neg A_1(\vec{t}_1), \dots, \neg A_n(\vec{t}_n), \neg B_{i+1}(\vec{u}_{i+1}), \dots, \neg B_m(\vec{u}_m) \} \theta.$$

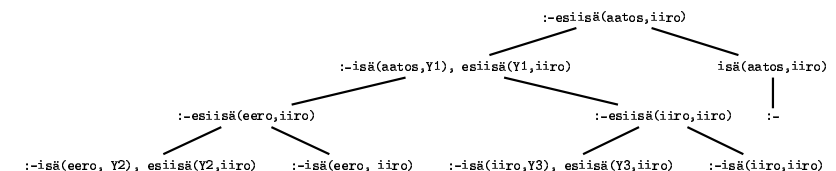
Esimerkki. Tarkastellaan seuraavaa PROLOG-ohjelmaa:

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:



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



9 Ohjelmien oikeellisuustarkastelut

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

9.1 Tarkasteltava ohjelmointikieli

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

Määritelmä. *Kokonaislukulausekkeet* E määritellään seuraavasti.

1. Mikä tahansa kokonaisluku $\dots, -1, 0, 1, \dots$ on kokonaislukulauseke.
2. Kokonaislukumuuttujat x, y, \dots 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.



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.

Määritelmä. *Boolean lausekkeet* B määritellään seuraavasti.

1. Boolean vakiot `false` ja `true` ovat Boolean lausekkeita.
2. Jos E_1 ja E_2 ovat kokonaislukulausekkeita, niin $E_1 > E_2$ on Boolean lauseke.
3. Jos B_1 ja B_2 ovat Boolean lausekkeita, niin negaatio $!B_1$, konjunktio $B_1 \ \&\& \ B_2$ ja disjunktio $B_1 \ || \ B_2$ ovat myös Boolean lausekkeita.
4. Muita Boolean lausekkeita ei ole.

Huomioita. Lausekkeet $E_1 == E_2$ (yhtäsuuruus), $E_1 != E_2$, $E_1 <= E_2$, $E_1 < E_2$ 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 $B_1 \rightarrow B_2$ 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 Boolean lauseke sekä C_1 ja C_2 ovat komentoja, niin myös

$C_1 ; C_2$	(ketjulause)
<code>if(B) then {C₁} else {C₂}</code>	(ehtolause)
<code>while(B) {C₁}</code>	(toistolause)

ovat komentoja.

3. Muita komentoja ei ole.

\Rightarrow *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 $x > 0$).

Huomioita.

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

Esimerkki. Tarkastellaan tilaa S , missä $x^S = 2$ ja $y^S = 6$ ja $z^S = 3$. Lausekkeiden $(x * x)$ ja $(z * z)$ arvot ovat $(x * x)^S = 4$ ja $(z * z)^S = 9$. Niinpä $S \models (x * x < y) \ \&\& \ (y < z * z)$, mutta $S \not\models (x * y < z)$.



Ohjelman tilojen esittäminen struktuurina

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

- (i) Struktuurin 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.

Komentojen suorittamisen vaikutus tilaan

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

1. Jos C on sijoituslause $x = E$, niin tila S' on $S[x \mapsto E^S]$.
2. Jos C on ketjulause $C_1 ; C_2$, niin S' on tila, joka saavutetaan suorittamalla ensin C_1 ja sitten C_2 .
3. Jos C on ehtolause `if(B) then {C1} else {C2}`, niin S' on tila, joka saavutetaan suorittamalla C_1 , jos $S \models B$, ja C_2 , jos $S \not\models B$.
4. Jos C on toistolause `while(B) {C1}` ja $S \not\models B$, niin $S' = S$.
5. Jos C on toistolause `while(B) {C1}` ja $S \models B$, niin S' on tila, joka saavutetaan suorittamalla $C_1 ; \text{while}(B) \{C_1\}$.



Huomioita.

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

Esimerkki. Tarkastellaan ohjelman

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

suoritusta tilasta S , missä $x^S = 3$. Ohjelman suorituksen aikana alkutilaa päivitetään seuraavasti:

$y \mapsto 1, z \mapsto 1, z \mapsto 2, y \mapsto 2, z \mapsto 3$ ja $y \mapsto 6$.

\implies muuttujan y arvona on muuttujan x arvon kertoma.

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

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.

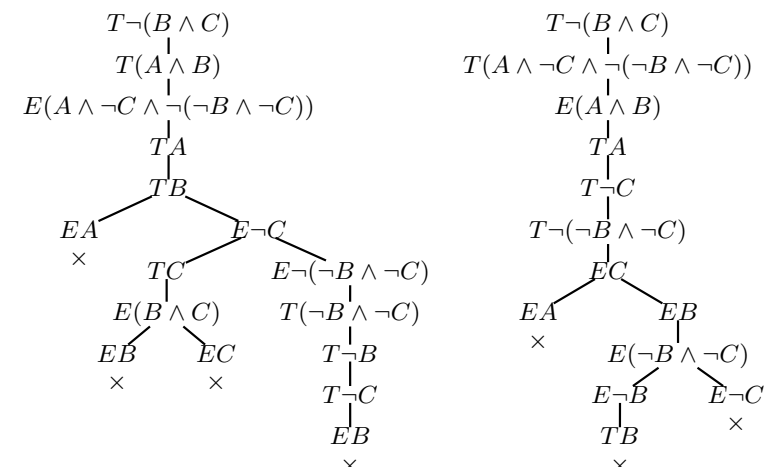


9.2 Boolean lausekkeiden ekvivalenssi

- Ohjelmointikielissä käytetään paljon ehtolauseita kontrolloimaan, millä ehdoilla ja mitä toimintoja suoritetaan.
- Jos ehtolauseita muutetaan esim. optimointitarkoituksessa, halutaan varmistua että toiminnot suoritetaan samoilla ehdoilla.
- Boolean lausekkeiden ekvivalenssin osoittamiseen voidaan käyttää sekä lauselogiikan että predikaattilogiikan menetelmiä.
- Jos Boolean 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. (Jatkoa) ekvivalenssi voidaan todeta semanttisella taululla:



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



Kytkenät predikaattilogiikkaan

- Boolean lauseke B on \mathbb{Z} -*pätevä* (merk. $\models_{\mathbb{Z}} B$) $\iff S \models B$ kaikissa \mathbb{Z} -struktuureissa S .
- Näin lausekkeiden muuttujat saavat universaalien tulkinnan.
- Boolean 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. $\models_{\mathbb{Z}} !((x > y) \ \&\& \ (y > x))$, mutta $\not\models !((x > y) \ \&\& \ (y > x))$, koska löytyy vastamalli S , missä $U = \{0\}$, $x^S = y^S = 0$ ja $>^S = \{(0, 0)\}$.

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

Osittainen ja täysi oikeellisuus

Olkoon P ohjelma sekä B_1 ja B_2 kaksi Boolean lauseketta.

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

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

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

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



9.3 Ohjelmien esi- ja jälkiehdot

- Tarkasteltavan ohjelmointikielen ohjelmilla on ääretön tila-avaruus, jonka läpikäyminen on käytännössä mahdotonta.
- Yksi mahdollisuus on tarkastella Boolean lausekkeiden B määrittelemiä tilajoukkoja $\{S \mid S \models B\}$ 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 $[B_1] P [B_2]$.
- Karkeasti ottaen ajatuksena on, että esiehdon B_1 on tarkoitus taata jälkiehdon B_2 voimaantulo ohjelman P suorituksen päättyessä.

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

Päättyvä sääntöjä osittaiselle oikeellisuudelle

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

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

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

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

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

Implikaatio:
$$\frac{\models_{\mathbb{Z}} B_1 \rightarrow B_2 \quad [B_2] \ C \ [B_3] \quad \models_{\mathbb{Z}} B_3 \rightarrow B_4}{[B_1] \ C \ [B_4]}$$

**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 askelella hyödynnetään Boolean lausekkeiden $(x == n) \ \&\& \ (y == m)$ ja $(y == m) \ \&\& \ (x == n)$ välistä ekvivalenssia.

Huomioita. Todistus voitaisiin kirjoittaa myös puun muotoon.

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

1. Jos B on jälkiehto sijoituslauseelle $x = E$, heikoimmaksi esiehtoksi voidaan kirjata $B\{x/E\}$ eli $\models_p [B\{x/E\}] \ x = E [B]$.

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 ($\models_z B_0 \rightarrow B_1$), 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 $\text{if}(B) \ \text{then}\ \{C_1\} \ \text{else}\ \{C_2\}$ heikoimmaksi esiehdoksi kirjataan $(B \ \&\& \ B_1) \ || \ (!B \ \&\& \ B_2)$.

Esimerkki. $[(x > y)] [((x > y) \ \&\& \ (x == y)) \ || \ (!(x > y) \ \&\& \ (y == y))]$
 $\text{if}(x > y) \ \text{then}\ \{x = x\} \ \text{else}\ \{x = y\} [x == y]$

**Heikoimmat esiehdot**

- Olkoon P ohjelma $C_1 ; \dots ; C_n$, missä C_1, \dots, 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, \dots, B_{n-1} siten, että $\models_p [B_{i-1}] \ C_i [B_i]$ on osoitettavissa kaikille $i \in \{1, \dots, n\}$.
- Usein tällaiset ehdot voidaan löytää analysoimalla komentosekvenssiä takaperin: haetaan komennolle C_i (missä i saa arvot $n, n-1, \dots, 1$) *heikoin esiehto* B_{i-1} siten, että $\models_p [B_{i-1}] \ C_i [B_i]$.

- Jatkoksa tällaisia todistuksia kirjoitetaan sekvensseiksi

$$[B_0] \ C_1 [B_1] \ C_2 [B_2] \ \dots \ C_n [B_n],$$

vaikka käytännössä sekvenssi muodostetaankin usein takaperin.

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.

```
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 $\text{while}(B) \{C\}$ invariantti I on mikä tahansa Boolean 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.

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

```
[true] [0 == 0 * y] z = 0 [0 == z * y] v = 0 [v == z * y] (A4)
```

```
while(!(x == z)) {
```

```
  [(v == z * y) && !(x == z)] (A3)
```

```
  [v + y == (z + 1) * y] z = z + 1 [v + y == z * y] v = v + y [v == z * y] (A2)
```

```
}
```

```
[(v == z * y) && (x == z)] [v == x * y] (A1)
```

Huomio. Todistuksessa käytetty invariantti on $v == z * y$.

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

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



Sopivan invariantin hakemisesta

- Osittaisen oikeellisuuden $\models_p [B_1] \ \text{while}(B) \ \{C\} \ [B_2]$ todistaminen voi perustua sopivan invariantin I käyttöön:

1. $\models_Z B_1 \rightarrow I$,

2. $\models_Z (I \ \&\& \ !B) \rightarrow B_2$, ja

3. $\models_p [I] \ \text{while}(B) \ \{C\} \ [I \ \&\& \ !B]$.

- Kuten aiemminkin, oikeellisuustodistusta voi hakea takaperin:

A1. Valitaan I siten, että $\models_Z (I \ \&\& \ !B) \rightarrow B_2$.

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

A3. Osoitetaan $\models_Z I \ \&\& \ B \rightarrow I'$, minkä nojalla $\models_p [I \ \&\& \ B] \ C \ [I]$ ja edelleen $\models_p [I] \ \text{while}(B) \ \{C\} \ [I \ \&\& \ !B]$.

A4. Osoitetaan $\models_Z B_1 \rightarrow I$.

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

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

missä E on sopiva kokonaislukulauseke, n on (uusi) kokonaislukumuuttuja ja B_1 on vahvennettu invariantti $B \ \&\& \ (0 \leq E)$.

- Näin lausekkeen E arvo pienenee jatkuvasti toistettaessa C :tä.
 \Rightarrow Suoritus päättyy vääjäämättä, koska $0 \leq E$ säilyy voimassa.



Esimerkki. Osoitetaan $\models_t [0 \leq x] \text{Multi } [v == x * y]$ 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 $0 \leq x$ on yllä välttämätön, koska $\not\models_t [\text{true}] \text{Multi } [v == x * y]!$