1 PREDIKAATTILOGIIKKA

1. Predikaattilogiikan kieli
2. Predikaattilogiikan semantiikka
3. Normaalimuodot
4. Semanttiset taulut predikaattilogikalle
5. Tietämysten esittämisestä
6. Herbrandin teoreema
7. Unifikaatio
8. Resolutiosääntö ja todistukset
9. Ohjelmien oikeellisuustarkastelut

1.1 Motivaatio

- Lauseologiikka on usein tarkoituksin liian yksinkertainen: olkoon
  $A = \text{"a on viallinen"}$, $B = \text{"b on viallinen"}$, $C = \text{"c on viallinen"}$.
  Tällöin kaikki ovat viallisia $= A \land B \land C$ ja
  "jokin on viallinen" $= A \lor B \lor C$.

- 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."

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

$C_d \land D_e \rightarrow C_e \land (C_c \land E_d \rightarrow C_d) \land (D_e \land E_c \rightarrow D_e) \land "$

Esimerkki. Alla on kuvattu joitain henkilöiden välisiä suhteita.

© 2004 TKK / Tietojenkäsittelyteorian laboratorio
1.2 Predikaattilogiikan aakkosto

Predikaattilogiikan kielessä $L$ käytetään seuraavia symboleja:
- Muuttujasymbolit $\mathcal{V} = \{x, y, z, \ldots\}$
- Vakiosymbolit $\mathcal{C} = \{a, b, c, \ldots\}$
- Funktionsymbolit $\mathcal{F} = \{f, g, h, \ldots\}$
- Predikaattisymbolit $\mathcal{P} = \{P, Q, R, \ldots\}$
- Lauseologian konnektiit $\neg, \land, \lor, \rightarrow, \leftrightarrow$
- Kvanttorisymbolit $\exists, \forall$
- Sulut ( ) ja pilku ,

1.3 Kielen määritelmä

Predikaattilogiikan kielen $L$ 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 $f \in \mathcal{F}$ on $n$-paikainen funktionsymboli ja $t_1, \ldots, t_n$ ovat termejä, niin myös $f(t_1, \ldots, t_n)$ on termi.
4. Muita termejä ei ole.

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

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

1. Jokainen atomikävenä φ on kaava.

2. Jos φ ja ψ ovat kaavoja ja x on muuttuja, niin myös

  (¬φ), (φ ∨ ψ), (φ ∧ ψ), (φ → ψ), (φ ↔ ψ), (∀xφ), (∃xφ)

  ovat kaavoja.

3. Muita kaavoja ei ole.

Esimerkki. Predikaattilogian kaavoja: P(c), (∀x(P(x) → Q(x))),

(∀x(P(x) ∨ (∃yQ(x,y)))), (∃x∀y(∀zP(x,y,z))).

Symbolijoukkoihin Ψ, C, F ja P perustava predikaattilogian kielik määrittelee edellä annetuilla periaatteilla muodostettavissa olevien kaavojen joukkona.

---

Kaavojen jäsennyspuut

Predikaattilogian kaavojilla on yksikäsitteinen jäsennyspuu.

Kaava ∀x(P(x) → Q(f(x,c)))

jäsennyspuu on annettu oikealla,

Kayserinen kaava on muodoltaan universalisti kvantifiointi kaava.

Huomio. Jäsennyspuun juuressa oleva konnektiivi määrittää edelleen, mitä muoto lause on, Esimerkiksi ∃xP(x) → ∀xP(x) on muodoltaan implikaatio, kun taas ∃x(P(x) → ∀xQ(y)) on muodoltaan existenceaalisesti kvantifiointi kaava.

---

Sopimukset sulkeiden käytöstä

- Konnektiivien presedenssiluokat ovat seuraavat:

  1. ¬, ∀v ja ∃v (missä v ∈ Ψ) ovat vahvimmat konnektiivit,

  2. ∨ ja ∧ ovat nältä heikompia, mutta vahvempia kuin → ja ↔,

  3. → ja ↔ ovat heikoimmat konnektiivit.

- Lauseologian yhteydessä käytettävät periaatteet sulkeiden vähentämiseksi käytettäen myös kaavoja kirjoitettaessa.

Esimerkki. Tätä kaava

(∃x∀y(∃z(P(x,z) ∧ P(z,y)))) → ((Q(x) ∨ Q(y)) ∨ R(x,y)))

voidaan kirjoittaa selkeämmin

∃x∀y(∃z(P(x,z) ∧ P(z,y)) → Q(x) ∨ Q(y) ∨ R(x,y)).

---

1.4 Kvanttoireihin liittyviä määritteleitä

Kaavan alkaavat määrittelevät seuraavasti:

- Atomisen kaavan ψ ainoa alkaava on ψ itse.

- Kaavan ∃ψ (₉ψψ) alkaavat ovat ∃ψ (₉ψψ) ja ψ:n alkaavat.

- Lauseologian konnektiivit (¬, →, ↔, ∨, ∧) käsitellään vastaavalla tavalla (vrt. alilausoiden määritelmä lauseologian tapauksessa).

Esimerkki. Kaava φ = ∃x(A(x) ∧ ∃y(S(x) ∩ N(x))) alkaavat ovat

φ, ∃x(A(x), ∃y(S(x) ∩ N(x)), A(x), S(x) ∧ N(x), S(x) ja N(x).
Vapaat ja sidotut muuttujat kaavassa

Määritelmä. Olkoot $\exists x \psi$ ja $\forall x \exists y$ predikaattilogiikan kaavojia. Alikäävä $\psi$ on kvanttorin $\exists x$ vaikutusalue kaavassa $\exists x \psi$. Vastaavasti alikaava $\phi$ on kvanttorin $\forall x$ vaikutusalue kaavassa $\forall x \phi$.

Esimerkki.

$\forall x \exists y (P(x) \rightarrow \exists y Q(x,y))$

$\exists x \exists y (Q(x) \leftrightarrow \forall z P(x,z)) \forall x \exists y R(x)$

Muuttujattoman termi sijoittaminen kaavaan

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

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

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

1.5 Lauseiden muodostaminen

1. Tunnistetaan kuvattavaa järjestelmään liittyvät objektit:
   - Oteka käyttöön vakiomäärittelyä jotka seuraavat käytössä.
   - Määritelmiä, joiden ohella oletettavaa lähettää oletettiin käytettävissä.

2. Tarkista, että suurin lauseen välissä on ja oletka käyttöön antaa käytettävissä.
Esimerkki. Olkoon $t$ = ”tuoli”, $h$ = ”hattu”, $s$ = ”sateenvarjo” vakiointa ja $P(x, y) = "x$ on y:n päällä,€ ”kaksipaikainen predikaatti. Tällöin:

$P(s, t) =$ ”sateenvarjo ei ole tuolin päällä”.

$\exists xP(x, h) =$

”on olemassa $x$, joka on hatun päällä”,
ei ”hatun päällä on jotakin”.

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

”on olemassa $x$ siten, että mikään $y$ ei ole $x:n$ päällä”,
ei ”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”,
ei ”Kaikki hatun päällä olevat ovat tuolin päällä”.

Esimerkki. Funktiosymbolit tarjoavat tavan esittää induktiivisia tietorakenteita termien avulla,

1. Luonnolliset luvut: vakiobsymboli 0 ja funktiosymboli $s \in F_1$,
   • Termit $0, s(0), s(s(0)), \ldots$ vastaavat luonnollisia lukujen 0, 1, 2, ...
2. Listat: vakiobsymbol $e$ (tyhjä lista) ja funktiosymbol $c \in F_2$,
   • Termit $e, c(a, e), c(a, c(b, e)), \ldots$ vastaavat listojen [...] $[a], [a, b], \ldots$
3. Binääripuitut: funktiosymbol $l \in F_1$ (lehtisoluut) ja $t \in F_2$
   (sisäsolmut),
   • Termit $l(a), l(l(a), l(b)), l(l(a), l(l(b), l(c))), \ldots$ vastaavat puitua

Esimerkki. Kuvataan henkilöiden vallitse suhtetta predikaattilogikalla,

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

$\exists x\forall y(Y(x, y) \land Y(y, x))$

$\exists x\forall y(S(x) \land S(y) \land V(x, y))$

$\exists xA(x, c) \land \exists x(S(x) \land N(x))$

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

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

Esimerkki. Esitettänyt binääripuitut kuten edellä funktiosymbolien $l$ ja $t$
avoilla, kirjoitetaan määritelmä seuraavalla predikaattille:

$P(x, y) =$ ”binääripuu $x$ on binääripuun $y$ pelikuva”.

Koska binääripuitut muodostavat induktiivisen tietorakenteen, on
luontevaa, että predikaattille $P(x, y)$ joudutaan kirjoittamaan
induktiiivinen/rekursiivinen määritelmä seuraavalla tavalla:

• Perustepaus (pelkästä lehtisolmusta koostuvat binääripuitut):
  $\forall x(P(l(x), l(x))).$

• Induktiiviesi (monimitkaesemmat binääripuitut):
  $\forall x\forall y\forall v(P(x, y) \land P(z, v) \rightarrow P(t(x, z), t(y, v))).$

⇒ Määritelmä kattaa kaikki binääripuitut.
2 Predikaattilogiikan semantikka

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

Huomioita.
- Joukon $U^n = \overbrace{U \times \cdots \times U}^{n \text{ kpl}}$ alkiot ovat monikkoja (tai jonoja) $\langle a_1, \ldots, a_n \rangle$, missä alkiot $a_1 \in U, \ldots, a_n \in U$.
- 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 alkiointiin. Muuttujan $v$ arvon valitminen struktuurissa $S$ tapahtuu seuraavalla tavalla,

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

2.1 Struktuurit

Predikaattilogiikassa toisuusjakalet korvataan struktuureilla,

Määritelmä. Struktuuri (rakenne) $S$ kielelle $L$ koostuu
- universumista $U$, joka on jokin ei-tyhjä joukko, sekä
- vakio-, muuttuja-, funktio- ja predikaattisymbolien tulkinnoota:
  1. Vakiosymbolin $c \in L$ tulkintana on alkio $c^S \in U$.
  2. Muuttujasymbolin $v \in V'$ tulkintana on alkio $v^S \in U$.
  3. Funktiosymbolin $f \in F_n$ tulkintana on funktio $f^S : U^n \rightarrow U$.
  4. Predikaattisymbolin $P \in P_n$ tulkintana on relatio $P^S \subseteq U^n$.

Struktuuri voidaan edelleen ymmärtää yhden asiointilain kuvauskensa,

Esimerkki.

- Helsinki$^S = \{he, tu, ha, be, lo\}$
- Tukholma$^S = \{tu\}$
- Berliini$^S = \{be\}$
- Lontoo$^S = \{lo\}$
- Paikkaapunkti$^S = \{he, tu, ha, be, lo\} \subseteq U^1 = U$
- Lento$^S = \{\langle he, tu, ha, be, lo \rangle, \langle tu, ha, be, lo \rangle, \langle ha, be, lo \rangle \} \subseteq U^2$
2.2 Pedikaattilogiikan totuusmäärittelmä

Termien tulkinta struktuurissa

Määrittelmä. Olkoon $S$ struktuuri kielelle $L$ ja $U$ struktuurin $S$ universumi.

- Vakio $c \in C$ nimeää universumin $U$ alkion $c^S$.
- Muuttuja $v \in V$ nimeää universumin $U$ alkion $v^S$.
- Jos termi $t_1, \ldots, t_n$ nimevät universumin $U$ alkiot $t_1^S, \ldots, t_n^S$ ja $f \in F_n$ niin termi $f(t_1, \ldots, t_n)$ nimeää universumin $U$ alkion $f^S(t_1^S, \ldots, t_n^S)$.

Näin voimme viittata kielen $L$ termeillä universumin $U$ alkioihin (kunhan vakio-, muuttuja- ja funktiosymbolien tulkinnat on annettu).

Esimerkki. Tarkastellaan vakiosymbolia $c \in C$ ja funktiosymbolia $f \in F_2$ ja $g \in F_1$. Olkoon struktuurin $S$ universumin $U$ luonnollisten lukujen joukko $0, 1, 2, \ldots$. Valitakem, symbolien tulkinnat seuraavasti:

- $c^S = 0$.
- $f^S = \seuraaja$, eli $f^S(n) = n + 1$, ja
- $g^S = \summa$, eli $g^S(n, m) = n + m$.

Tällöin $c$ nimeää alkion 0,

$\begin{align*}
 f(c) & \text{ nimeää alkion 1,} \\
 f^n(c) & = f(f(\ldots f(c)\ldots)) \text{ nimeää alkion } n \text{ ja } n \text{ kpl} \\
 g(f(c), f(f(c))) & \text{ nimeää alkion 3.}
\end{align*}$

Kaavojen totuusarvojen laskeminen struktuurissa

Olkoon $S$ struktuuri kielelle $L$ ja $U$ struktuurin $S$ universumi.

Määrittelmä. Seurauksina määrittelmän, milloin kaava $\phi \in L$ on toti struktuurissa $S$ (merk. $S \models \phi$) ja milloin epätoti (merk. $S \not\models \phi$).

1. $S \models t_1 = t_2 \iff t_1^S$ ja $t_2^S$ ovat sama universumin $U$ alkio (yllä $t_1$ ja $t_2$ ovat mitä tahansa termiä),

2. $S \models P(t_1, \ldots, t_n) \iff$

\[
\langle t_1, \ldots, t_n \rangle^S \text{ (eli jono } \langle t_1^S, \ldots, t_n^S \rangle \text{) kuuluu tulkintaan } P^S
\]

(yllä $n > 0$, $P \in F_n$ ja $t_1, \ldots, t_n$ ovat mitä tahansa termiä),

3. $S \not\models P \iff$ tyhjä jono () kuuluu tulkintaan $P^S$ (missä $P \in F_0$),

4. $S \models \neg \alpha \iff S \not\models \alpha$,

5. $S \models \alpha \land \beta \iff S \models \alpha$ ja $S \models \beta$.

6. $S \models \alpha \lor \beta \iff S \models \alpha$ tai $S \models \beta$.

7. $S \models \alpha \rightarrow \beta \iff S \not\models \alpha$ tai $S \models \beta$.

8. $S \not\models \alpha \leftrightarrow \beta \iff$ joko $S \models \alpha$ ja $S \models \beta$, tai $S \not\models \alpha$ ja $S \not\models \beta$.

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

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

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

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

Väite. Kaaville kaavioille $\phi \in L$ pätee joko $S \models \phi$ tai $S \not\models \phi$.

Väite. Jos kaava $\phi \in L$ on lisäksi lause, sen totuusarvo ei riipu muuttujien $v \in V$ tulkinnosta struktuurissa $S$. 

© 2004 TKK / Tietojenäyttelyteorian laboratorio
**Esimerkki.** Tarkastellaan graafin sähmujen joukkoa (universumi) ja esitetään kaaret kaksipaikaisen predikaatin $K$ avulla. Nyt esim. graafia

$$
\begin{array}{c}
s_0 \\
\downarrow
\end{array}
\begin{array}{c}
s_1 \\
\downarrow
\end{array}
\begin{array}{c}
s_2
\end{array}
$$

vasta struktuuri $S$, jonka universumi on $U = \{s_0, s_1, s_2\}$ ja $K$-n tulkinta

$$K^S = \{\langle s_0, s_1 \rangle, \langle s_1, s_2 \rangle, \langle s_2, s_2 \rangle\}$$

(muuttujat $x$ ja $y$ tulkitaan vapaasti).

1. $\langle s_0, s_1 \rangle \in K^S$
   $\implies$ $\langle x, y \rangle \in K^{s_0+s_1}
   \implies S[x \mapsto s_0, y \mapsto s_1] = K(x, y)$
   $\implies S[x \mapsto s_0] = \exists y K(x, y)$

2. Vastaavasti $\langle s_1, s_2 \rangle \in K^S$
   $\implies S[x \mapsto s_1] = \exists y K(x, y)$

3. Vastaavasti $\langle s_2, s_2 \rangle \in K^S$
   $\implies S[x \mapsto s_2] = \exists y K(x, y)$

---

**Esimerkki.** (jatkoka)

4. Koska $S[x \mapsto s_1] = \exists y K(x, y)$ jokaiselle universumin alkioille $s_1 \in U$, voimme todeta, että $S = \forall x \exists y K(x, y)$.

5. Lisäksi esim.
   
   $$\langle s_2, s_2 \rangle \in K^S$$
   $\implies$ $\langle x, x \rangle \in K^{s_2+s_2}$
   $\implies S[x \mapsto s_2] = K(x, x)$
   $\implies S[x \mapsto s_2] \models \neg K(x, x)$
   $\implies S \models \forall x \neg K(x, x)$
   $\implies S \models \neg \forall x K(x, x)$

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

---

**2.3 Semanttiset peruskäsitteet**

- Semanttisten peruskäsitteiden määritelmät ovat muodoltaan samat.
- Olennaisena erona lauseologiakan on, että lauseiden rakenne on monipuolisempi ja että struktuurit korvavat totuusjälkeen.

**Määritelmä.** Struktuuri $S$ on lauseen $\alpha \in L_{mali}$, jos lause $\alpha$ on toisi struktuurissa $S$ eli $S \models \alpha$.

**Määritelmä.** Struktuuri $S$ on lausejoukon $\Sigma \subseteq L_{mali}$, jos kaikille lausejoukon $\Sigma$ lauseille $\sigma \in \Sigma$ pätee $S \models \sigma$. 
Määritelmä. Lause $\alpha \in \mathcal{L}$ (tai lausejoukko $\Sigma \subseteq \mathcal{L}$) on **toteutuva**, jos ainakin yksi struktuuri $S$ on sen malli.

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

1. $(1,1) \in P^S$ $\implies$ $(x,y)_{\sigma = 1, \sigma = 1} \in P(x,y).
\implies S[x \mapsto 1, y \mapsto 1] \models P(x,y)$.
2. $(1,2) \in P^S$ $\implies$ $(x,y)_{\sigma = 1, \sigma = 2} \in P(x,y).
\implies S[x \mapsto 1, y \mapsto 2] \models P(x,y)$.
3. Sis $S[x \mapsto 1] \models \forall y P(x,y)$ ja $S \models \exists x \forall y P(x,y)$.

© 2004 TKK / Tietojenkäsittelyteorian laboratorio

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

**Esimerkki.** Osoitetaan $\models \forall x P(x) \leftrightarrow \neg \exists x \neg P(x)$.
Kaikille kielen $\mathcal{L}$ struktuureille $S$ ja vastaaville universumille $U$ pätee:

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

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

© 2004 TKK / Tietojenkäsittelyteorian laboratorio

2.4 **Vastamallit**

Vastamallin käsittää 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 vastamall on struktuuri $S$ siten, että $S \not\models \phi$.
- Jos $\Sigma \models \phi$, niin vastamall on struktuuri $S$ siten, että $S \models \Sigma$ ja $S \models \phi$.

**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) $S$:

$U = \{1,2\}$, $A^S = \emptyset$, ja $B^S = \{1\}$.
Tätä $S \models \forall x (A(x) \rightarrow B(x))$, $S \models \forall y \neg A(y)$, $S \models \forall y A(y) \vee \forall y \neg A(y)$,
$S \not\models \forall z B(z)$, $S \not\models \forall z \neg B(z)$ ja edelleen $S \not\models \forall z B(z) \vee \forall z \neg B(z)$.
(Muuttujien $x$, $y$ ja $z$ tulkinnat voikaan valita vaapasti struktuurissa $S$).
Esimerkki. Luokitellaan viikonpäiviä seuraavalla lausejoukolla:

$$\Sigma = \{ \text{tiistai}, \ \text{keskiviikko}, \ \text{luento} \}$$

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

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

\[
\begin{align*}
\text{tiistai}^S & = t, \\
\text{keskiviikko}^S & = k, \\
\text{perjantai}^S & = p, \\
\text{luento}^S & = \{ k, p \},
\end{align*}
\]

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

2.5 Peruskäsitteiden väliset yhteydet

Seuraavat ominaisuudet ovat voimassa myös predikaattilogiassa:

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

Olko $$C_n(\Sigma) = \{ \phi \in \mathcal{L} \mid \phi \text{ on lause ja } \Sigma \models \phi \}$$ lausejoukolle $$\Sigma \subseteq \mathcal{L}$$:

- $$\Sigma \subseteq C_n(\Sigma)$$ ja $$\Sigma \equiv C_n(\Sigma)$$.
- Monotonisuus: $$\Sigma_1 \subseteq \Sigma_2 \implies C_n(\Sigma_1) \subseteq C_n(\Sigma_2)$$.
- $$C_n(C_n(\Sigma)) = C_n(\Sigma)$$.

3 Normaaliimuodot

- Prenex-normaaliimuoto
- Konjunktivinen normaaliimuoto
- Eksistenssikvanttorien eliminointi
- Lauseiden klausulimuoto predikaattilogiassa

3.1 Prenex-normaaliimuoto

Lause $$\alpha$$ on prenex normaaliimuodossa, 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 $$\phi$$ ei sisällä kvanttoreita.

Esimerkki. Seuraavat lauseet ovat prenex-normaaliimuodossa:

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

Välite. Jokainen predikaattilogiakan lause on logiisesti ekvivalentti jonkin prenex-normaaliimuodossa olevan lauseen kanssa.
Lauseiden muuttaminen prenex-normaalimuotoon

Mikä tahansa predikaattilogian lause voidaan muuttaa prenex normaalimuotoon seuraavalla menetellyllä:

1. Poistetaan konnektiivit → ja ↔:
   \[ \phi \rightarrow \psi \sim -\phi \lor \psi \]
   \[ \phi \leftrightarrow \psi \sim (-\phi \lor \psi) \land (\phi \lor -\psi) \]

2. Viedään negaatiot lauserakenteen sisään (atomisten kaavojen eteen):
   \[ -\phi \sim \phi \]
   \[ -\phi \lor \psi \sim -\phi \land \psi \]
   \[ -\phi \land \psi \sim -\phi \lor \psi \]

Yllä \( \forall x \) on mikä tahansa kvanttorien sekkumen.

Esimerkki. Muunnetaan seuraava lause prenex-normaalimuotoon:

\[
\forall x(P(x) \rightarrow \exists z R(z, x)) \rightarrow \exists x Q(x)
\]

\[
\sim -\forall x(P(x) \rightarrow \exists z R(z, x)) \lor \exists x Q(x)
\]

\[
\sim -\exists z\neg R(z, x) \lor \exists x Q(x)
\]

\[
\sim \exists x(P(x) \land \forall z \neg R(z, x)) \lor \exists x Q(x)
\]

\[
\sim \exists x(P(x) \land \forall z \neg R(z, x)) \lor Q(x)
\]

\[
\sim \exists x(P(x) \land \forall z \neg R(z, x)) \lor Q(x)
\]

\[
\sim \exists x(P(x) \land \forall z \neg R(z, x)) \lor Q(x)
\]

\[
\sim \exists x(P(x) \land \forall z \neg R(z, x)) \lor Q(x)
\]

3. Tuodaan kvanttorit ulos lauserakenteesta:

\[
\forall x(\forall y(\psi(y) \lor \psi)) \sim \forall x \forall z(\phi(z) \lor \psi)
\]

\[
\forall x(\forall y(\psi(y) \land \psi)) \sim \forall x \forall z(\phi(z) \land \psi)
\]

\[
\forall x(\psi(y) \lor \forall y(\psi(y))) \sim \forall x \forall z(\psi(y) \lor \phi(z))
\]

\[
\forall x(\psi(y) \land \forall y(\psi(y))) \sim \forall x \forall z(\psi(y) \land \phi(z))
\]

Yllä \( \forall y \) korvataan uudella muuttujalla \( z \), mikäli \( y \) esiintyy vapaana alikaavassa \( \psi \). Muussa tapauksessa \( x \) voi olla kyseinen eli. \( z \)

Eksistenssikvanttorit \( \exists y \) käsitetään samaan tapaan (saadaan 4 vastaavanmuotoista sääntöjä lisäämällä).

Esimerkki. Muuttujan korvaaminen uudella on olemassa:

\[
\forall x P(x) \lor \exists x Q(x) \sim \forall x (P(x) \lor \exists x Q(x)) \sim \forall y (P(y) \lor Q(y)).
\]

3.2 Konjunktiivinen normaalimuoto

Määritelmä. \( \text{Literaali} \) ovat joko

1. atomiakaavojen \( P(\overline{t}) \) (eli \( \text{positiivisaa literaalaa} \)) tai

2. atomiakaavojen negaatioita \( \neg P(\overline{t}) \) (eli \( \text{negatiivisaa literaalaa} \)).

Yllä \( \overline{t} \) tarkoittaa termien \( t_1, \ldots, t_n \) sekkuvien kun \( P \in \mathcal{T} \), jos \( P \in \mathcal{T} \), sekkuvien on tyhjä ja tällöin myös sulkujen sisällön kirjoittamatta näkyvän.

Määritelmä. Lause \( \alpha \) on konjunktiivisessa normaalimuodossa, mikäli se on

1. prenex-normaalimuodossa \( \forall x_1 \exists x_2 \ldots \exists x_n \phi \), missä kvanttoreita sisältämätön osa \( \phi = \phi_1 \land \cdots \land \phi_n \) ja jokainen konjunktion jäsen \( \phi_i \) on

Esimerkki. Seuraava lause on konjunktiivisessa normaalimuodossa:

\[
\forall x \exists y \forall z ((-P(x, y) \lor Q(y, x)) \land R(z) \land (-R(x) \lor P(y, z) \lor Q(x, z))).
\]
3.3 Eksistenssinkvanttorien eliminointi

**Esimerki.** Tarkastellaan kahta kokonaissatuksia:

1. Summafunktioilla on yleisimpiä:
   \[ \exists x \forall y(x + y = y). \]
   Identiteettialkio voikaa nimetä vakiomuodolla 0:
   \[ \forall y(0 + y = y). \]
2. Jokaista kokonaissatuksia on vastaluku:
   \[ \forall x \exists y(x + y = 0), \]
   Vastalukufunktio voidaan nimetä funktiosymbolilla \(-\):
   \[ \forall x(x + -x = 0). \]
Skolemoinnin logiset ominaisuudet

**Väite.** Prenex-normaalimuodon oleva lause \( \varphi \) on toteutuva \( \iff \) lauseen \( \varphi \) skolemoitu muoto \( \varphi' \) on toteutuva.

**Huomio.** Prenex normaalimuodon olevan lauseen \( \varphi \) skolemoitun muotot \( \varphi' \) ei välttämättä ole logiisesti ekvivalentti lauseen \( \varphi \) kanssa.

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

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

Vastamalli \( S \): universumi \( U = \{1,2\} \), \( e^S = 1 \) ja \( P^S = \{2\} \).

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

Tätä myös \( \not \models \exists x P(x) \leftrightarrow P(c) \) ja edelleen \( \exists x P(x) \not \models P(c) \).

---

3.4 **Lauseiden klausulimuoto predikaattilogiikassa**

Mille tahansa lauseelle voidaan hakea klausulimuoto seuraavasti:

1. Haetaan prenex-normaalimuoto,
3. Tarvittaessa poistetaan eksistensikvanttorit skolemoimalla,

**Esimerkki.** Klausulitiesys lauseelle \( \forall x (\neg P(x) \to \forall y Q(x,y)) \lor R(x) \):

\[
\begin{align*}
\models & \forall x \exists z ((P(x) \land \neg Q(x,z)) \lor R(x)) \quad (1) \\
\models & \forall x \exists z ((P(x) \lor R(x)) \land (\neg Q(x,z) \lor R(x))) \quad (2) \\
\models & \forall x ((P(x) \lor R(x)) \land (\neg Q(x,f(x)) \lor R(x))) \quad (3) \\
\models & \{P(x) \lor R(x), \{\neg Q(x,f(x)), R(x)\} \}. \quad (4)
\end{align*}
\]

---

4. **Semanttiset taulut predikaattilogiikalle**

- Taulusäännöt kvanttoideiden käsittelyyn
- Semanttiset taulut predikaattilogiikalle
- Ohjeita taulutodistusten laadintaan
- Systemaattinen taulu
- Vastamallin konstruointi
4.1 Taulusäännöt kvanttorien käsittelyyn

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

\[
\begin{array}{c|c}
T \exists \forall \varphi(x) & E \exists \forall \varphi(x) \\
\hline
T \varphi(c) & E \varphi(c) \\
\hline
c \ uusi vakiot & c \ uusi vakiot
\end{array}
\]

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

_Huomio._ Uuden vakiannon johduu siitä, ettemme tiedä, millä universumin alkiolla on ko. ominaisuus $\varphi$ (tai ei ole ominaisuutta $\varphi$).

---

4.2 Semanttiset taulut predikaattilogikalle

- Semanttisten taulujen määritelmiä säilyy ennallaan.

- Rajaamme yhtäsuuruispredikaatin "=" tarkastelun ulkopuolelle välttääksemme yhtälöiden käsittelyn (vrt. *equality axioms*).

- Ehtoja, millä polun solmu on *hajoittu*, joudutaan täydentämään: Olkoon $\tau$ semanttinen taulu ja $P$ polku juurisolmusta lehtisolmuun tassa. $P$-n solmu $T \exists \forall \varphi(x)$ ($E \exists \forall \varphi(x)$) hajoittuu polulla $P$, jos

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

_Huomio._ Mikäli polulla $P$ ei esiinny vakiosymboleita, $T \exists \forall \varphi(x)$ ($E \exists \forall \varphi(x)$) tulee hajoittaa käyttäen jotain uutta vakiosymbolia c.
Esimerkki. Allaolevan semanttisen taulun kaikki polut ovat valmiit:

1. \( T \forall x(P(x) \rightarrow Q(x)) \)
2. \( E(Q(a) \lor Q(b)) \)
3. \( E\{Q(a) \} \)
4. \( E\{Q(a) \} \)
5. \( T(P(a) \rightarrow Q(a)) \) \(1, x/a \)
6. \( EP(a) \) \(5 \)
7. \( T(P(b) \rightarrow Q(b)) \) \(1, x/b \)
8. \( EP(b) \) \(7 \)

Taulu on ristiriitainen. Lause \( Q(a) \) on siis johdettavissa lausejoukosta \( \{P(a), \forall x(P(x) \rightarrow Q(x))\} \) ja siten myös lausejoukon loginen seurauks.

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

1. \( T \forall x(P(x) \rightarrow Q(x)) \)
2. \( TP(a) \)
3. \( E\{Q(a) \} \)
4. \( T(P(a) \rightarrow Q(a)) \) \(1, x/a \)
5. \( EP(a) \) \(4 \)
6. \( TQ(a) \) \(4 \)

Taulu on ristiriitainen. Lause \( Q(a) \) on siis johdettavissa lausejoukosta \( \{P(a), \forall x(P(x) \rightarrow Q(x))\} \) ja siten myös lausejoukon loginen seurauks.
4.3 Ohjeita taulutodistusten laadintaan

- Lauseen rakenne määrää edelleen, mitä tauluosiäntöä tulee käyttää (jäsennyspuun juuressa oleva konnektiivi).

- Solmuja hajoittaminen myösannottaa voin vaikuttaa taulun kokoon (haaratumaista kannattaa vältttää).

- Jälkimäsennettä quantorisoimalla termin y tilalle valitaa hajoittamishetkellä (esim. ymehennin) jokin vakio tai funktio- ja vakiosymboleista rakentaa muuttujat tilatiem.

- Valitsena muuttujatotamon termit y sopivasti voidaan usein nopeuttaa taulun valmistumista.

---

2. Solmu \( \forall x \phi(x) \) (tai \( E \exists x \phi(x) \)) joudutaan hajoittamaan useasti.

**Esimerkk.** \( \{\forall x \phi(x)\} \vdash P(a) \land P(b) \)

1. \( \forall x \phi(x) \) \hfill 1. \( \forall x \phi(x) \)
2. \( E(P(a) \land P(b)) \hfill E(P(a) \land P(b))
3. \( TP(a)^1, x^a \leftrightarrow \hfill TP(a)^1, x^a \leftrightarrow \)
4. \( TP(b)^1, x^b \leftrightarrow \hfill TP(b)^1, x^b \leftrightarrow \)
5. \( EP(a)^2 \leftrightarrow \hfill EP(a)^2 \leftrightarrow \)
6. \( EP(b)^2 \leftrightarrow \hfill EP(b)^2 \leftrightarrow \)

---

3. Muuttujen korvaaminen sopivilla muuttujatomilla termeillä.

**Esimerkk.** \( \{\forall x \forall y \exists z (P(x,y) \land P(y,z) \rightarrow P(x,z)), P(a,b), P(b,c)\} \vdash P(a,c) \)

- Semanttiseen taulun tulee solmu \( \forall x \forall y \exists z (P(x,y) \land P(y,z) \rightarrow P(x,z)) \), josta voidaan johtaa kvanttorisäännöllä 27 erilaista tautoksi merkittyä implikaatioita.

- Ristiriidan johtamisen kannalta dennaasia ovat implikaatioista ne, joissa esiintyy atomisia lauseita \( P(a,b), P(b,c) \) ja \( P(a,c) \).

- Esimerkin tapauksessa tämä johtaa ensimmäiseksi \( x \), y:n ja \( z:n \) korvaamiseen vakioilla \( a, b \) ja \( c \) (näin saatava implikaatio riittää).

- Muita implikaatioita ei tarvita, ja niiden johtaminen ja mahdollinen hajoittaminen johtaa semanttisen taulun tarpeettoman kasvuun.

---

"Taulutodistusten erityispiirteitä pedikaattilologiikan tapauksessa"

1. Valitaan muuttujatomaksi termiksi y vakio, joka ei esiinny lauseessa.

**Esimerkk.** Esitettäen \( \{\forall x \phi(x)\} \vdash \exists x \phi(x) \).

1. \( \forall x \phi(x) \) \hfill 2. \( \exists x \phi(x) \)
3. \( TP(c)^1, x^c \leftrightarrow \hfill TP(c)^1, x^c \leftrightarrow \)
4. \( EP(c)^2, x^c \leftrightarrow \hfill EP(c)^2, x^c \leftrightarrow \)

**Huomio.** Tämä on perusteltua, koska universumissa \( U \) on ainakin vähintään yksi alkio \( a \in U \), joka voidaan nimetä (eli \( c^x = a \)).
Kvanttorisekvenssin käsitteily

Jatkossa sallimme seuraavien johdettujen taulussäintöjen käytön kvanttorisekvenssien käsitteleyssä:

\[
\begin{array}{c|c}
T \forall a \forall y \forall z (P(x, y) \land P(y, z) \rightarrow P(x, z)) \\
2, TP(a, b) \\
3, TP(b, c) \\
4, EP(a, c) \\
5, T \forall a \forall y \forall z (P(a, y) \land P(y, z) \rightarrow P(a, z))^1, x/a \\
6, T \forall z (P(a, b) \land P(b, z) \rightarrow P(a, z))^5, y/b \\
7, T (P(a, b) \land P(b, c) \rightarrow P(a, c))^5, z/c \\
8, E (P(a, b) \lor P(b, c))^7, 8, TP(a, c)^7, \\
9, EP(a, b)^8, 9, EP(b, c)^8, \\
\end{array}
\]

Yllä \( c_1, \ldots, c_n \) ovat \( a \)-, taulussäintöjen edellyttämiä usia vakioida ja vastaavasti \( t_1, \ldots, t_n \) ovat valittuja muistutajottia termejä.

4.4 Systemaattinen taulu

- Lauseologian keskeiset päätelyongelmat ovat ratkeavia.

\textbf{Esimerkki}. Voidaan konstruoida \textit{determinininen} Turing kone \( T \), jonka suoritus pysähtyy syotteeksi annettua lauseologian lauseella \( \varphi \):

1. hyväksymään tilaan \( k \) (kyllä), jos syöte \( \varphi \) on pätevä, ja
2. hyväksymään tilaan \( e \) (ei), jos syöte \( \varphi \) ei ole pätevä.

\textbf{Huomioita}.

- Tällainen algoritmi voi perustua esim. totuusaulukkoihin, semantiiksiin tauluihin tai resoluutioon.
- Myös looginen ekvivalenttisuus, looginen seuraavuus ja toteutuvuus ovat lauseologian tapauksessa ratkeavia ongelmaa.
Esimerkki. Lauseen φ pätevyyden tarkastamista varten voidaan konstruoida seuraavanlainen deterministinen Turing kone T:
1. Jos syöte φ on pätevä, T pysähtyy hyväksyvään tilaan k (kylkä).
2. Jos syöte φ ei ole pätevä, T pysähtyy joskus hylkäävään tilaan e (ei) ja joskus T ei pysähdy lainkaan.

Huomio. Tälläkin algoritmi voi perustua semanttisiin tauluihin:
- Rakentamalla semanttinen taulu tietyställä tavalla systemaattisesti, voidaan taata, että taulu saadaan aina ristiriitaisiksi, kun sen juuressa on Eφ ja φ on pätevä.

Systemaattisen taulun periaatteita
- Tuotetaan indeksoinnalla riittävän määrä uusia vakoitia c1,c2,c3,..., kun hajoitetaan muotoa T\forallψ(x) tai E\existsψ(x) olevia soluja,
- Tuotetaan tarpeen mukaan muuttujattomia termiä t1,t2,t3,..., jotka rakentuvat Eψ:ssä esiintyvistä vakoijaa ja funktiosymboliista sekä mahdollisestä käyttöönotetuista uusista vakoista c1,c2,c3,...
- Sekvenssin t1,t2,t3,... on oltava reilu: jokainen em. symboleista rakentuva muuttujaton termi on esiintynyyn siinä jonakin termin t1.
- Hajoitusten reilut: taataan, että taulun keskeneräisillä poluilla esiintyvät hajoittamattomat solmut tulevat halutussuoroon (seuraavan kerran) äärellisen monen muun hajoituksen jälkeen,
- Muoto T\forallψ(x) tai E\existsψ(x) olevia soluja hajoitetaan järjestysessä käyttäen muuttujattomia termiä t1,t2,t3,...
Esimerki. Käytetään reilua hajoitusjärjestystä:

1. \( T \forall x G(s(x),x) \)
2. \( T \forall x \forall y G(x,y) \rightarrow G(s(x),y) \)
3. \( E G(s(s(0)), s(0)) \)
4. \( T G(s(0),0) \)
5. \( T \forall y (G(0,y) \rightarrow G(s(0),y)) \)
6. \( T G(0,0) \rightarrow G(s(0),0) \)
7. \( E G(0,0) \)
8. \( T G(s(0),s(0)) \)
9. \( T \forall y (G(s(0),y) \rightarrow G(s(0),y)) \)

Systemaattinen taulu voi tehdä turhaa työtä \( \Rightarrow \) heuristikka tarvitaan!

4.5 Vastamallien konstruointi

- Vastamallin (struktuuri) \( S \) konstruoimisessa voidaan hyödyntää semanttisen taulun ristiriidattomasta polusta saatavia \textit{atomisia lauseita} koskevia totuusarvovaatimuksia \( TP(t_1, \ldots, t_n) \), \( EQ(s_1, \ldots, s_m) \), \( \langle t_i \rangle \) ja \( s_j \) ovat muuttujattomia termejä.
- Valtaan riittävän iso universumi \( U \), jotta pystytään antamaan tulkinnat totuusarvovaatimuksissa esintyville vakio- ja funktiosymboleille.
- Tämän jälkeen valitaan predikaattien tulkinnat totuusarvovaatimusten mukaan:
  1. Jos \( TP(t_1, \ldots, t_n) \) on polulla, \( \langle t_1^S, \ldots, t_n^S \rangle \in P^S \).
  2. Jos \( EQ(s_1, \ldots, s_m) \) on polulla, \( \langle s_1^S, \ldots, s_m^S \rangle \notin Q^S \).

Esimerki. Valitsemalla muuttujattomat termit aikaisemmin esitettyllä periaatteilla semanttinen taulu jää huomattavasti pienemmäksi:

1. \( T \forall x G(s(x),x) \)
2. \( T \forall x \forall y (G(x,y) \rightarrow G(s(x),y)) \)
3. \( E G(s(s(0)), s(0)) \)
4. \( T \forall y (G(s(s(0)), s(0)) \rightarrow G(s(s(0)),s(0))) \)
5. \( T G(s(s(0)),s(0)) \rightarrow G(s(s(0)),s(0)) \)
6. \( E G(s(s(0)),s(0)) \)
7. \( T G(s(s(0)),s(0)) \)

Esimerki. Vastamallilla \( S \) lauseen \( \forall x (P(x) \rightarrow Q(x)) \) pätevyydelle:

1. \( E \forall x (P(x) \rightarrow Q(x)) \)
2. \( E (P(c) \rightarrow Q(c)) \)
3. \( T (P(c)) \)
4. \( E (Q(c)) \)

1. Totuusarvovaatimukset ristiriidattomasta polusta: \( TP(c) \) ja \( EQ(c) \).
2. Räätää, että universumiin \( U = \{ 1 \} \) otetaan yksi alkio s.e. \( c^S = 1 \).
3. Totuusarvovaatimusten nojalla: \( 1 \in P^S \) ja \( 1 \notin Q^S \).
4. Nämä vaatimukset toteutuvat valinnoilla \( P^S = \{ 1 \} \) ja \( Q^S = \emptyset \).
Esimerkki. \( \{ \forall x (P(x) \land Q(x) \rightarrow R(x)) \} \nRightarrow \forall x (Q(x) \rightarrow R(x)) \),

1. \( T \forall x (P(x) \land Q(x) \rightarrow R(x)) \)
2. \( E \forall x (Q(x) \rightarrow R(x)) \)
3. \( E (Q(c) \rightarrow R(c)) \)
4. \( T Q(c) \)
5. \( E R(c) \)
6. \( T (P(c) \land Q(c) \rightarrow R(c)) \)
7. \( E (P(c) \land Q(c)) \)
8. \( E P(c) \)
8. \( E Q(c) \)

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

1. \( E \exists x (P(a) \lor P(f(x))) \)
2. \( E (P(a) \lor P(f(a))) \)
3. \( E P(a) \)
4. \( E P(f(a)) \)
5. \( E P(a) \lor P(f(a)) \)
6. \( E P(f(a)) \)
7. \( E P(f(f(a))) \)

Edellytyksenä on muuttujattomien termien tulkinta samalle \( U \)n alkiolle.

Tarkasteillaan taulun ainoa ristiriidatonta polkuja \( P \),
- Polulla esiintyy yksi vakiosymboli \( c \) muttei funktiosymbolia,
- Voidaan muodostaa ainoastaan yksi muuttujaton termi eli \( c \) itse,
- Täten solmu \( T \forall x (P(x) \land Q(x) \rightarrow R(x)) \) on hajoitettu polulla \( P \),
  koska polulla \( P \) on solmu \( T (P(t) \land Q(t) \rightarrow R(t)) \) jokaista muuttujatonta termiä \( t \in \{ c \} \) kohtaan.
- Polku \( P \) on siis valmis,
- Nämä ollen taulu on kokonaisuutena myös valmis,
- Polulta \( P \) saadaan totuusarvavaatimukset \( E P(c) \), \( T Q(c) \) ja \( E R(c) \).
- Muodostetaan vastamallin \( S \) valitsemalla universumiksi \( U = \{ 1 \} \) ja symbolien tulkinnoksi \( c^S = 1 \), \( P^S = R^S = 0 \) ja \( Q^S = \{ 1 \} \).
5 Tietämyksen esittämisestä

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

5.1 Tietämyksen esittäminen predikaattilogiikalla

Annettuun järjestelmään liittyvää tietämystä voidaan esittää valitsemalla
• sopiva predikaattilogiikan aakkosto (joukot \( P, C \) ja \( f \)) ja
• vastaavan kielen \( L \) perustuva lausejoukko \( \Sigma \subseteq L \), jonka lauseet määrittelevät järjestelmän ominaisuudet.

Tarkastellaan määritelmää vastaavan lausejoukon \( \Sigma \subseteq L \) logisten seurausten joukkoa \( \text{CN}(\Sigma) = \{ \phi \in L \mid \phi \text{ on lause ja } \Sigma \models \phi \} \). Nyt
• \( \Sigma \) muodostaa järjestelmää koskevan eksplisiittisen tietämyksen ja
• joukon \( \text{CN}(\Sigma) \) se lauseet ovat implisiittistä tietämystä eli väittämiä, jotka voidaan päätellä eksplisiittisestä tietämyksestä.

Esimerkki. Kuvataan radioverkon linkkien välityksellä syntyviä yhteyksiä seuraavalla predikaattilogiikan lausejoukolla \( \Sigma \):

\[
\{ \text{linkki}(a,b), \text{linkki}(b,c), \text{linkki}(d,e), \\
\forall x \text{ yhteydy}(x,x), \\
\forall x \forall y (\text{linkki}(x,y) \rightarrow \text{yhteydy}(x,y)), \\
\forall x \forall y (\text{yhteydy}(x,y) \rightarrow \text{yhteydy}(y,x)), \\
\forall x \forall y (\text{yhteydy}(x,y) \land \text{yhteydy}(y,z) \rightarrow \text{yhteydy}(x,z)) \}.
\]

Nyt esim., lause \( \text{linkki}(a,b) \) on eksplisiittistä (ylöskirjattua) tietämystä, kun taas esim., lause \( \text{yhteydy}(c,a) \) lukeutuu lausejoukon \( \Sigma \) logisena seurausena osaksi implisiittistä tietämystä.
Esimerkki. Palauttaa radioverkkoesimerkin lausejoukkoon $\Sigma$, jonka osalta voidaan todeella 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$^\mathcal{S} = \{(1, 2), (2, 3), (4, 5)\}$ ja
  
  - yhteys$^\mathcal{S} = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 4), (4, 5), (5, 4), (5, 5)\}$

- 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.

5.2 Ohjeita predikaattien määrittelemiseen

- Tavoitteena kirjoittaa annetulle predikaattille $P$ (ja siten myös sen
  tulkinnan olevalle relaatiolle) määritelmä joidenkin muiden
  predikaattien avulla.

- Mieliivaltainen predikaattilogian kaava $\phi$ voidaan saattaa muotoon

  $Q_1x_1Q_2x_2\ldots Q_nx_n\psi$

  missä kokin kvanttori $Q_i$ on joko $\forall$ tai $\exists$, ja kaava $\psi$ on
  konjunktiivisessa normalimuodossa eikä sisällä kvanttoriteita.

- Yllä $\psi = \psi_1 \wedge \ldots \wedge \psi_m$, missä kokin $\psi_i$ on literaalien disjunktio

  $\neg Q_1(\bar{t}_1) \lor \ldots \lor \neg Q_n(\bar{t}_n) \lor P_1(\bar{s}) \lor \ldots \lor P_l(\bar{s})$

  $\equiv Q_1(\bar{t}_1) \wedge \ldots \wedge Q_n(\bar{t}_n) \rightarrow P_1(\bar{s}) \lor \ldots \lor P_l(\bar{s})$.

Määritelmiä tarkkuudesta

- Olkoon $H$ tarkasteltavan predikaattilogian kielen
  muuttujatomen termien joukko.

- Tavoitteemenne on siis kirjoittaa predikaatin $P \in T_H$ määrittelevä
  lausejoukko $\Sigma_P$, kun lähtökohtana on tieto predikaatin $P$
  tarkoittamasta relaatiosta $P^* \subseteq H^n$.

- Määritelmä $\Sigma_P$ voidaan pitää räätälän tapana, jos kaikille
  muuttujatommille termeille $t \in H$, $\bar{t}_n \in H$ pätee seuraavaa:

  $\langle t_1, \ldots, t_n \rangle \in P^* \iff \Sigma_P \models P(t_1, \ldots, t_n)$.

- Tälläinen positivistinen määritelmä ei ole välttämättä täydellinen eli
  määritelmä ei taka, että $\Sigma_P \models P(t_1, \ldots, t_n)$ mikäli $\langle t_1, \ldots, t_n \rangle \not\in P^*$.
**Esimerki.** 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)), $ $\forall x\forall y\forall z(\text{yhteys}(x,y) \land \text{yhteys}(y,z) \rightarrow \text{yhteys}(x,z)) \}$.  

Määritelmän lähtökohtana on relaatio $\text{linkki}^* = \{(a, b), (b, c), (d, e)\}$. Yhteys-predikaatin määritelmä voidaan pitää riittävän tarkan, koska tavoiteltu relaatio $\text{yhteys}^* = \{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c), (c, a), (c, b), (c, c), (d, d), (d, e), (e, d), (e, e)\}$ saadaan määritelmän $\Sigma$ loogisena seurauskseen edellä kuvattuella tavalla. Määritelmä ei ole täydellinen, koska esim. $\Sigma \models \neg \text{yhteys}(a,e)$.

---

**Esimerki.** Olkoon annettuna predikaatti  
1. sairastaa(x) = ”henkilö x on sairaas” ja  
2. tapaa(x,y) = ”henkilö x tapaa henkilön y”. 

Tarkoituksena on määritellä näiden avulla predikaatti tartuntavaarassa(x) = ”henkilö x on tartuntavaarassa.” 
Kysymys: millä ehdolla jonkin henkilön on tartuntavaarassa?
1. Jos henkilö tapaa jonkun sairaan henkilön.
2. Jos henkilö tapaa jonkun toisen tartuntavaarassa olevan henkilön.

Yritetään kirjoittaa nämä edellä esitetyn mukaisesti muotoon  
$\forall x \forall y \ldots (Q_1(\vec{i}) \land \ldots \land Q_k(\vec{i}) \rightarrow \text{tartuntavaarassa}(x)).$

---

**Määritelmien käyttö konkreettisessa päätelyssä**

**Esimerki.** Lisätään edellä johdettuun tartuntavaarassa-predikaatin määritelmän tietokanta, jossa kuvataan tapaamiset ja sairastamiset:  
Näin saadaan lausejoukko  
$\Sigma = \{ \forall x\forall y(\text{tapaa}(x,y) \land \text{sairastaa}(y) \rightarrow \text{tartuntavaarassa}(x)), $ $\forall x\forall y(\text{tapaa}(x,y) \land \text{tartuntavaarassa}(y) \rightarrow \text{tartuntavaarassa}(x)), $ $\forall x\forall y(\text{tapaa}(x,y) \rightarrow \text{tapaa}(y,x)), $ $\text{tapaa}(\text{Lyyli}, \text{Hemmo}), \text{tapaa}(\text{Lyyli}, \text{Erkkii}), \text{sairastaa}(\text{Erkkii}) \}$.  

Kysymys asetelmassa saadaan  
$\Sigma \models \text{tartuntavaarassa}(\text{Lyyli}) \land \text{tartuntavaarassa}(\text{Hemmo})$.  
Kokeile tämän osoittamista semanttisella taululla!
**Tyypitettä kvanttoitä**

- Usein on mielekästä ajatella universumin koostuvan tyypitään erilaisista alkioista.
- Tällöin syntyy tarve rajata kvantifiointia koskemaan ainoastaan tiettyä tyyppeä $T$ devia alkioita seuraavan tapan:
  \[ \forall x \in T : \phi(x) \text{ ja } \exists x \in T : \phi(x). \]
- Tyypit $T$ voidaan esittää yksipäikäisen predikaatin avulla:
  \[ T(x) = \text{"alkio } x \text{ on tyyppeä } T\". \]
- Tyypitettä kvanttoitka ilmaistaan predikaattilogikassa seuraavasti:
  \[ \forall x(T(x) \to \phi(x)) \text{ ja } \exists x(T(x) \land \phi(x)). \]

**Esimerkki.** Lisätään edellisen esimerkkiin tyypippedikaatteja.

- Määritellään predikaatit henkilöiden ja tautien erottelemiseksi:
  henkilö($x$) = "$x$ on henkilö" ja tauti($x$) = "$x$ on tauti".
- Määritellään predikaatit ilman tyyppeinformaatiota:
  - tapaa($x, y$) = "$x$ tapaa $y:n",
  - sairastaa($x, y$) = "$x$ sairastaa $y$:tä" ja
  - tartuntavaarassa($x, y$) = "$x$ on varassa sairastua $y$:n".
- Lauseet saadaan nyt seuraavaan muotoon:
  \[ \forall x \forall y \forall z (\text{henkilö}(x) \land \text{henkilö}(y) \land \text{tapa}(x, y) \land \text{tauti}(z) \land \text{sairastaa}(x, y) \to \text{tartuntavaarassa}(x, z)) \text{ ja} \]
  \[ \forall x \forall y \forall z (\text{henkilö}(x) \land \text{henkilö}(y) \land \text{tapa}(x, y) \land \text{tauti}(z) \land \text{tartuntavaarassa}(y, z) \to \text{tartuntavaarassa}(x, z)). \]

**Muodotaan monimutkaisempia määritelmiä**

- Edellä otettiin lähtökohtaksi muotoa
  \[ \forall x_1 \forall x_2 \cdots \forall x_n (Q_1(\vec{t}) \land \cdots \land Q_k(\vec{t}) \to P(\vec{t})) \]
  olevat määritelmiä. Näiden ilmaisuvuonna ei ole aina riittävä.
- Joissakin tilanteissa tarvitaan eksistentiaalista kvantifiointia:
  \[ \forall x (\text{solmu}(x) \to \exists y (\text{väri}(y) \land \text{väristety}(x, y))) \]
  \[ \equiv \forall x \exists y (\text{solmu}(x) \to \text{väri}(y) \land \text{väristety}(x, y)). \]
- Implikaation seurauksena voi olla myös atomien disjunktio
  \[ P_1(\vec{s}) \lor \cdots \lor P_n(\vec{s}) \] pelkän atomin $P(\vec{t})$ sijaan:
  \[ \forall x (\text{bitti}(x) \to \text{nolla}(x) \lor \text{yksi}(x)). \]

**Huomio.** Edellä oli keskeistä vaihtoehtoisuuden ilmaiseminen,
5.3 Nimien yksikäsitteisyys ja kattavuus

- Rajoitetaan jatkossa pedikaattilogiikan kielin $L$, jossa ei ole funktiosymboleita ja ainoastaan ääreillä määrrä vakiosymboleita.
- Pedikaattilogiikassa struktuurin $S$ määritelmä ja tapa jolla vakiosymbolit tulkitaan $S$:ssa mahdollistavat, että
  1. jokin universumin $U$ alkio $a \in U$ on useammien vakiointien \( c_1, \ldots, c_n \)\, ($n > 1$) nimeämä: $c_1^a = \ldots = c_n^a = a$.
  2. jokin universumin $U$ alkio $a \in U$ ei ole minkään vakiointien nimeämä (eli kaikille vakiosymboleille $c$ pätee $c^a \neq a$).
- Tietämyksen esittämisen kannalta tällainen mahdollisuus muodostuu usein jopa turhaksi, mutta:

Nimeämien voidaan pakottaa yksikäsitteiseksi lauseita lisäämällä.

### Esimerkki

Tarkastellaan lausejoukon

$$ \Sigma_{UNA} = \{ \neg (\text{Lyyli} = \text{Hemmo}), \neg (\text{Lyyli} = \text{Erkki}), \neg (\text{Hemmo} = \text{Erkki}) \} $$
malleja $S_n$ kun universumina $U_t$ on joukko henkilöitä $h_1, h_2, \ldots$.

<table>
<thead>
<tr>
<th>$U_t$</th>
<th>Lyyli$^S$</th>
<th>Hemmo$^S$</th>
<th>Erkki$^S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${ h_1, h_2, h_3 }$</td>
<td>$h_1$</td>
<td>$h_2$</td>
<td>$h_3$</td>
</tr>
<tr>
<td>${ h_1, h_2, h_3 }$</td>
<td>$h_1$</td>
<td>$h_3$</td>
<td>$h_2$</td>
</tr>
<tr>
<td>${ h_1, h_2, h_3 }$</td>
<td>$h_2$</td>
<td>$h_1$</td>
<td>$h_3$</td>
</tr>
<tr>
<td>${ h_1, h_2, h_3 }$</td>
<td>$h_3$</td>
<td>$h_2$</td>
<td>$h_1$</td>
</tr>
<tr>
<td>${ h_1, h_2, h_3, h_4 }$</td>
<td>$h_1$</td>
<td>$h_2$</td>
<td>$h_3$</td>
</tr>
</tbody>
</table>

$\implies$ Universumissa oltaa vähintään 3 henkilöä.

### Nimien yksikäsitteisyys

- Vastaava käsite englanniksi on unique names assumption (UNA).
- Kun kielellä on ääreellinen määrrä vakiosymboleita $c_1, \ldots, c_n$ riittää lisätä muotoa

  $$ \neg (c_i = c_j) $$

  olevat lauseet, missä $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n\}$ ja $i < j$.
- Lauseita tarvitaan neljällinen määrrä (yhteensä $2^{2^4}$ kappaletta).

### Esimerkki

Olkoon kielellä $L$ 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 kielellä on ääreellinen määrrä vakiosymboleita $c_1, \ldots, c_n$ riittää lisätä seuraavaa muotoa oleva lause:

  $$ \forall x (c_1 \lor \cdots \lor c_n = x) $$

- Tarvittavat lauseen pituus riippuu lineaarisesti vakiooiden lukumäärästä $n$.

### Esimerkki

Edellisen esimerkkin mukaisessa kielellä tarvitaan lause

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

© 2004 TKK / Tietojenkäsittelyteorian laboratorio
**Esimerkki.** Tarkastellaan lausejoukon 

\[ \Sigma_{\text{DCA}} = \{ \forall x (Lyli x = \text{Hemmo} \lor x = \text{Erkki}) \} \]

malleja \( S \), kun universumina \( U \) on joukko henkilöitä \( h_1, h_2, \ldots \).

\[
\begin{array}{ccc}
U_i & Lyli^i & \text{Hemmo}^i & \text{Erkki}^i \\
\{h_1, h_2, h_3\} & h_1 & h_2 & h_3 \\
\{h_1, h_2, h_3\} & h_1 & h_3 & h_2 \\
\{h_1, h_2, h_3\} & h_2 & h_3 & h_1 \\
\{h_1, h_2, h_3\} & h_3 & h_2 & h_1 \\
\end{array}
\]

\[ \implies \text{Universumissa voi olla korkeintaan 3 henkilöä.} \]

**Yhtäsuuruuspredikaatin määritelmä**

Jos yhtäsuuruuspredikaattia sisältävää lausea käytetään määritelmissä, seuraavat aksiomat saattavat olla tarpeen esim, todistuksissa.

1. **Reflaksivisyys:** \( \forall x (x = x) \).
2. **Symmetrisisyys:** \( \forall x \forall y ((x = y) \rightarrow (y = x)) \).
3. **Transitivisuus:** \( \forall x \forall y \forall z ((x = y) \land (y = z) \rightarrow (x = z)) \).
4. **Sijoittauvuus** (kaikille predikaateille \( P \in \mathcal{P}_a \)):

\[
\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n \\
(P(x_1, \ldots, x_n) \land (x_1 = y_1) \land \ldots \land (x_n = y_n) \rightarrow P(y_1, \ldots, y_n)).
\]

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

**Esimerkki.** Tarkastellaan vielä edeltävien lausejoukkojen unionin 

\[ \Sigma_{\text{UNA}} \cup \Sigma_{\text{DCA}} = \{ \neg (Lyli = \text{Hemmo}), \neg (Lyli = \text{Erkki}), \\
\neg (\text{Hemmo} = \text{Erkki}), \\
\forall x (Lyli x \lor x = \text{Hemmo} \lor x = \text{Erkki}) \} \]

malleja \( S \), kun universumina \( U \) on joukko henkilöitä \( h_1, h_2, \ldots \).

\[
\begin{array}{ccc}
U_i & Lyli^i & \text{Hemmo}^i & \text{Erkki}^i \\
\{h_1, h_2, h_3\} & h_1 & h_2 & h_3 \\
\{h_1, h_2, h_3\} & h_1 & h_3 & h_2 \\
\{h_1, h_2, h_3\} & h_2 & h_3 & h_1 \\
\{h_1, h_2, h_3\} & h_3 & h_2 & h_1 \\
\end{array}
\]

\[ \implies \text{Universumissa on oltava täsmälleen 3 henkilöä.} \]
Määritelmien täydellisyys

Määritelmä. Predikaattien \( P \in \mathcal{P}_n \) määritelmä \( \Sigma \subseteq L \) on täydellinen, jos
\[ \Sigma_P \models P(t_1, \ldots, t_n) \text{ tai } \Sigma_P \not\models P(t_1, \ldots, t_n), \]
kaikille kielen \( L \) muuttujattomille termeille \( t_1, \ldots, t_n \).

Huomioita.

• Jos predikaattien \( P \in \mathcal{P}_n \) määritelmä \( \Sigma_P \) on ristiriitainen (eli sillä ei ole malleja), se on trivialisesti täydellinen: tällöin sekä
\[ \Sigma_P \models P(t_1, \ldots, t_n) \text{ että } \Sigma_P \not\models P(t_1, \ldots, t_n) \]
kaikille muuttujattomille termeille \( t_1, \ldots, t_n \in L \).

• Jos predikaattien \( P \in \mathcal{P}_n \) määritelmä \( \Sigma_P \) on täydellinen ja
\[ \Sigma_P \not\models P(t_1, \ldots, t_n) \]
jollekin muuttujattomille termeille
\( t_1, \ldots, t_n \in H \), niin \( \Sigma_P \) ei ole ristiriitainen ja \( \Sigma_P \not\models P(t_1, \ldots, t_n) \).

Esimerkki. Tarkastellaan muunnelmaa tartuntavaara-keskustelua:
\[ \Sigma = \{ \forall x (\text{tapaa}(x, y) \land \text{sairastaa}(y) \rightarrow \text{tartuntavaaraa}(x)) \}, \]
\[ \forall x (\neg \text{sairastaa}(x) 
\land \neg \text{tartuntavaaraa}(x) \rightarrow \text{turvassa}(x)) \}, \]
\[ \text{tapaa}(\text{Lyly, Hemmo}), \text{tapaa}(\text{Lyly, Erkki}), \text{sairastaa}(\text{Erkki}) \}. \]

• Nyt \( \Sigma \models \text{tartuntavaaraa}(\text{Lyly}), \Sigma \not\models \text{tartuntavaaraa}(\text{Hemmo}) \) ja
\( \Sigma \not\models \neg \text{tartuntavaaraa}(\text{Hemmo}) \).

• Täten \( \Sigma \) ei ole täydellinen määritelmä tartuntavaaraa-predikaattien.

• Jotta näin olisi, määritelmästä tulisi seuraaja loogisesti
\( \neg \text{tartuntavaaraa}(\text{Hemmo}) \) ja \( \neg \text{tartuntavaaraa}(\text{Erkki}) \).

• Kyseisen määritelmän \( \Sigma \) ei ole myöskään täydellinen muille ai, kielen predikaateille (tapa, sairastaa, turvassa ja =).
Esimerkki. Tarkastellaan vastaavaa konstruktiota lausejoukolle

\[ \Sigma = \{ \text{tuntee1(Lyl), tuntee1(Hemmo, Hemmo),} \]
\[ \forall x \forall y (\text{tuntee1}(x, y) \rightarrow \text{tuntee2}(x, y)), \]
\[ \forall x \forall y (\text{tuntee2}(y, x) \rightarrow \text{tuntee2}(x, y)) \}. \]

- Rekursiivisesti määritellyn predikaatin tuntee2 taikoituksena on täydentää predikaattijoukon symmetriseksi.
- Täydennettynä määritelmä saadaan muotoon

\[ \Sigma' = \{ \neg (\text{Lyl} = \text{Hemmo}), \forall x (x = \text{Lyl} \lor x = \text{Hemmo}), \]
\[ \forall x \forall y (\text{tuntee1}(x, y) \leftrightarrow (x = \text{Lyl} \land y = \text{Lyl}) \lor (x = \text{Hemmo} \land y = \text{Hemmo})), \]
\[ \forall x \forall y (\text{tuntee2}(x, y) \leftrightarrow \text{tuntee1}(x, y) \lor \text{tuntee2}(x, y)) \}. \]

6 Herbrandin teoreema

- Herbrand-universumit
- Herbrand-struktuurijoukko
- Herbrandin teoreema
- Lause-ja predikaattilogiikan suhteesta

Yllä mainittujen täydennettyistä määrittelemistä ei seuraa logisesti-
\[-\text{tuntee2(Lyl, Hemmo)}\] eikä \[-\text{tuntee2(Hemmo, Lyl)}.\]

Lausejoukolla \( \Sigma' \) on seurava epäintuitiivinen malli \( S \):
Universumiksi \( U = \{ h_1, h_2 \} \),
\( \text{Lyl}^S = h_1 \), \( \text{Hemmo}^S = h_2 \),
\( \text{tuntee1}^S = \{ (h_1, h_1), (h_2, h_2) \} \) ja
\( \text{tuntee2}^S = \{ (h_1, h_1), (h_2, h_1), (h_2, h_2) \} \).

Kyseinen struktuuri \( S \) on vastamalli, koska
\( S \not\models \neg \text{tuntee2(Lyl, Hemmo)} \) ja \( S \not\models \neg \text{tuntee2(Hemmo, Lyl)} \).

Huomio. Tentissä ei edellytetä täydellisen määrittelemien kirjoittamista
predikaatteille (ellei tästä sitten erikseen jossain yksinkertaissessa
tapauksessa pyydetä).

6.1 Herbrand-universumit

Määritelmä. Predikaatit yllä laskeutuneen kielen \( \mathcal{L} \) Herbrand universumi \( H \) on
niiden muuttujapäätösten termien joukko, jotka ovat muodostettavissa
kielen \( \mathcal{L} \) vakiom- ja funktiosymboleista.

Esimerkki. Olkoon kelessä \( \mathcal{L} \) ainoastaan yksi vakiomsymboli \( c \) ja yksi
funktiosymboli \( f \in \mathcal{F}_2 \).
Herbrand universumiksi saadaan muuttujapäätösten termien joukko
\( H = \{ c, f(c, c), f(f(c, c), c), f(c, f(c, c)), f(f(c, c), f(c, c)), \ldots \} \).

Huomio. Jos kelessä \( \mathcal{L} \) ei ole funktiosymboleita ja ainoastaan äärellinen
määri vakiota, Herbrand-universumit \( H \) jää täällön äärelliseksi.
6.2 Herbrand-struktuurit ja -mallit

Määritelmä. Kielen \( \mathcal{L} \) Herbrand-struktuuri on struktuuri \( \mathcal{H} \), jonka
1. universumina on kielen \( \mathcal{L} \) Herbrand-universumit \( H \),
2. jokaisen vakiossymbolin \( c \in C \) tulkintana \( c^{\mathcal{H}} \) on \( c \) itse,
3. jokaisen funktiosymbolin \( f \in \mathcal{F} \) tulkintana on funktio \( f^{\mathcal{H}} \), joka
   kuvaa muuttujatottamen termin \( t_1, \ldots, t_n \) muuttujatottamaksi termiksi
   \( f(t_1, \ldots, t_n) \), ja
4. jokaisen predikaattisymbolin \( P \in \mathcal{P} \) tulkintana on \( P^{\mathcal{H}} \subseteq H \).

Jos lausejoukossa \( \Sigma \) ei esiinny yhtään vakiossymbolia,
Herbrand universumiin valitaan ainakin yksi vakiossymboli \( c \)
(strukturien määritelmän mukaan universumit ovat aina esihja).

Lausejoukon \( \Sigma \) Herbrand-universumista \( H \) on niiden
muuttujatottamien termien \( t \) joukko, jotka ovat muodostettavissa
lausejoukossa \( \Sigma \) esintytä vakiota ja funktiosymboloista.

Esimerkki. Lausejoukon \( \Sigma = \{ \forall x P(x, f(x)) \} \) Herbrand-universumi on
\( H = \{ c, f(c), f(f(c)), \ldots \} = \{ f^n(c) \mid n \geq 0 \} \).

Määritelmä. Kielen \( \mathcal{L} \) Herbrand-struktuuri \( \mathcal{H} \) on
1. lauseen \( \phi \in \mathcal{L} \) Herbrand-malli \( \iff \mathcal{H} \models \phi \), ja
2. lausejoukon \( \Sigma \subseteq \mathcal{L} \) Herbrand-malli \( \iff \mathcal{H} \models \sigma \) kaikille \( \sigma \in \Sigma \).

Esimerkki. Tarkastellaan lausejoukkoa
\( \Sigma = \{ P(a), \forall x (P(x) \rightarrow Q(x)), \forall x (Q(x) \rightarrow Q(f(x)) \land R(x, f(x))) \} \).

Lausejoukon \( \Sigma \) Herbrand-universumina on \( H = \{ f^n(a) \mid n \geq 0 \} \).

Muodostetaan Herbrand-struktuuri \( \mathcal{H} \), jonka universumina on \( H \) siten, että jokainen muuttujatossa termi \( t \in H \) tulkitaan \( t^{\mathcal{H}} = t \), ja
\( P^{\mathcal{H}} = \{ a \}, Q^{\mathcal{H}} = H \) ja \( R^{\mathcal{H}} = \{ (f^n(a), f^{m+1}(a)) \mid n \geq 0 \} \).

Kyseinen struktuuri \( \mathcal{H} \) on lausejoukon \( \Sigma \) Herbrand-malli.
6.3 Herbrandin teoreema

- Rajoitetaan tarkastelemaan klausulijoukkoa.
- Merkintä $C(x_1, \ldots, x_n)$ tarkoittaa muuttujat $x_1, \ldots, x_n$ sisältävää klausulia $\{P_1(\bar{t}), \ldots, P_k(\bar{t}), -Q_1(\bar{s}), \ldots, -Q_l(\bar{r})\}$.
- Kausuull $C(x_1, \ldots, x_n)$ vastaa universaalisti kvantifiointua lausetta $\forall x_1 \cdots \forall x_n \phi(C(x_1, \ldots, x_n))$, missä $\phi(x_1, \ldots, x_n)$ on klausuuli $C(x_1, \ldots, x_n)$ esitys literaalien disjunktioa.
- Klausulijoukkel $S$ voidaan määritellä Herbrand struktuurit samaan tapaan kuin lausejoukkoihin.
- Klausulijoukko $S$ voidaan *instantioida* vastaavan Herbrand-universumin $H_S$ suhteen seuraavasti:

6.4 Lauseologian ja predikaatilogian suhteesta

- Lauseogika on osa predikaatilogiikkaa:
  - Kaikki lauseologian konnektiivit ovat käytettävissä predikaatilogiikassa, 
  - 0-paikkaiset predikaatit vastaavat atomisia lauseita.
- Lauseologian päätelmät ja logiset ongelmat voidaan suorittaa/ratkota sellaisena predikaatilogiikan puiteessa.
- Herbrandin teoreeman nojalla predikaatilogian päätely voidaan palauttaa lauseologian päätelyksi.
- Lauseologian ja predikaatilogian *ilmastoinnassa* (eli kysymystä esittää) on kuitenkin huomattava ero.
Ilmaisuvoimaren ilmentymen:
- Áäreltä predikaattilogiikan lausejoukkoa saattaa vastata áäretön lauseologikan lausejoukko,
- Lauseologikan ratkeavuus vs. predikaattilogiikan puoliratkeavuus,

Rajoittamalla syntaksia sopivasti saadaan predikaattilogiikkallekin ratkeavia (ja ilmaisuvoimaaltaa heikompia) osajoukoja,
- Esim. jos S on áärellinen ja siinä ei esinny funktsiosymboleja, sen Herbrand-áinstanssien joukko S' jää áärelliseksi,
- Täällöin S:n toteutuvuus on selvitettyvissä áärellisessä ajassa.

Esimerkk. Klausuulijoukon $\{\{P(a), \neg P(x), P(b)\}\}$
Herbrand-universumi $H = \{a, b\}$ ja Herbrand-áinstanssien joukko $S' = \{\{P(a), \neg P(a), P(b)\}, \{\neg P(b), P(b)\}\}$, joka voidaan nähdä lauseologikan klausuulijoukkaa $S' = \{\{P\}, \{\neg P\}, \{\neg Q\}\}$.

7 Unifikaatio

- Substituutiot
- Yleimimat unifioijat
- Unifikaatiotaidegoritmi

7.1 Substituutiot

Määritelmä. Substituutio (tai korvau) $\theta$ on áärellinen joukko

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

missä $x_i$:t ovat muuttujia ja $t_i$:t korvauvia termejä siten, että
1. korvattavat muuttujat $x_1, \ldots, x_n$ ovat toisistaan eriytä ja
2. mikään korvavaa termi $t_i$ ei de muuttuja $x_i$ itse eli $t_i \neq x_i$.

Liäksi erotetaan seuraavat erikoistapaukset:

- Jos korvaavat termit $t_i$ ovat muuttujattomia, $\theta$ on muuttujaton,
- Jos korvaavat termit $t_i$ ovat muuttuja, $\theta$ on nimeämissubstituutio.

Esimerkki. Esimerkkeinä todettakoon

- tyhjä substituutio $\varepsilon = \{\}$,
- 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, atomikäava, literaali, klausuuli tms.) ja $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$ substituutio.

Lauseke $E\theta$ on muuton rakenteeltaan kuten $E$, paitsi että jokainen muuttujan $x_i$ esimittä lausekeessa $E$ on korvattu termillä $t_i$.

Jos lausekeessa $E\theta$ ei esiinny muuttuja, kutsutaan lauseet alla $E$ muuttujattomaksi instanssiksi.
### 7.2 Yleisimmät unifioijat

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

Lausekejoukon $S$ on unifioituva, mikäli sillä on ainakin yksi unifioija.

**Esimerkki.** Tarkastellaan seuraavien joukkojen unifioituuvuutta.

<table>
<thead>
<tr>
<th>Joukko $S$</th>
<th>Unifioija $\vartheta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${P(x, f(a)), P(y, z)}$</td>
<td>${x/y, z/f(a)}$ tai ${x/y, z/f(a)}$</td>
</tr>
<tr>
<td>${P(x, f(x)), P(x, f(x))}$</td>
<td>${x/f(x), y/f(f(a))}$</td>
</tr>
<tr>
<td>${P(a), P(f(x))}$</td>
<td>ei unifioija</td>
</tr>
<tr>
<td>${P(x), P(f(x))}$</td>
<td>ei unifioija (termit aina äärellisiä)</td>
</tr>
</tbody>
</table>

### 7.3 Unifikaatioalgoritmi

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

**Määritelmä.** Olkoon $S$ ei tyhjä joukko johonkin predikaattisymboliin $p$ perustuva atomikaava $\{P(\tilde{r}_1), \ldots, P(\tilde{r}_n)\}$.

1. Joukon $S$ eroakoa on vasemmalla oikealle siirryttäessä ensimmäinen kohta, jossa joukon $S$ atomikaavojen merkkijonoesityksissä on jokin eroavaa.$^2$

2. Joukon $S$ eroajoukko $D(S)$ kuuluvat atomikaavojen $P(\tilde{r}_1), \ldots, P(\tilde{r}_n)$ eroakohdasta halvat termit $u_1, \ldots, u_n$.

**Esimerkki.** Joukko $S_1 = \{P(x, a), P(x, y)\}$ eroajoukko $D(S_1) = \{a, y\}$.

Joukko $S_2 = \{Q(g(x, y), y), Q(g(x, f(z)), x), Q(g(x, f(a)), f(a))\}$ eroajoukko $D(S_2) = \{y, f(z), x\}$.
**Unikaatioalgoritmi** ehdottaa atomikaavojen joukolla $S$:

1. Jos joukon $S$ atomikaavojen predikaattiymbolit eivät ole samat, totea
   ettei $S$ unifioituja ja lopeta algoritmin suoritus.
2. Aseta $k := 0, S_k := S$ ja $s_k := \varepsilon$.
3. Jos $S_k$ on yksialkioinen (ja siten unifioiutunut) joukko, totea $S$
   unifioituvaaksi ja lopeta algoritmin suoritus.
4. Laske joukon $S_k$ eroajukko $D(S_k)$.
5. Jos $D(S_k)$:ssa on muuttuja $v_k$ ja termei $t_k$ on, että $v_k$ ei
   esiinny $t_k$:ssa, jatka algoritmin suoritusta kohdasta 7.
7. Aseta $s_k := v_k/t_k$ ja laske $S_{k+1} := S_k\{v_k/t_k\}$.
8. Aseta $k := k+1$ ja jatka algoritmin suorittamista kohdasta 3.

**Välte.** Olkoon $S$ ärellinen ehdottaa atomikaavojen.

- Jos $S$ on unifioituva, niin unikaatioalgoritmin suoritus päättyy
  askeleena 3 kohdalla ja substituutioiden $s_0, s_1, \ldots, s_k$ komposikio
  $\sigma = s_0 s_1 \cdots s_k$ on joukon $S$ yleisin unifioija
- Jos $S$ ei ole unifioituva, niin unikaatioalgoritmin laskenta päättyy
  askeleessa 1 tai askeleessa 6.

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

1. Predikaattiymbolit ovat samat, jatkaan.
2. $k = 0, S_0 = \{P(x, f(x)), P(g(a), z)\}, \sigma_0 = \varepsilon$.
3. $S_0$ ei ole yksialkioinen, jatkaan.

**Resoluutiosääntö ja -todistukset**

- Resoluutiosääntö predikaattilogian tapauksessa
- Resoluutiotodistukset
- Ohjeita resoluutiotodistusten kirjoittamiseen
- Tukijoukkostategia
8.1 Resoluutiosääntö predikaattilogiikan tapauksessa

Määritelmä. Olkoot

\[ C_1 = C'_1 \cup \{P(\bar{t}_1), \ldots, P(\bar{t}_m)\} \quad \text{ja} \quad C_2 = C'_2 \cup \{-P(\bar{u}_1), \ldots, -P(\bar{u}_m)\} \]

kaksi klausuulia,

1. joska ei esiinny yhteisää muuttuja ja
2. joska esiintyvien atomikaavojen joukko

\[ \{P(\bar{t}_1), \ldots, P(\bar{t}_m), P(\bar{u}_1), \ldots, P(\bar{u}_m)\} \]

on unifioidua (yleisimpänä unifioidujana \( \sigma \)),

Klausuulien \( C_1 \) ja \( C_2 \) yhdistelmä on klausuli \( C'_1 \sigma \cap C'_2 \sigma \).

Huomio. Yllä käytetty merkkintä \( A \cup B \) tarkoittaa keskenään alkioveraiden \( (A \cap B = \emptyset) \) joukkojen \( A \) ja \( B \) unionia \( A \cup B \).

Esimerkki. Tarkasteilla seuraavia klausuleja:

\[ C_1 = \{Q(x), -R(y), P(x,y), P(f(z), f(z))\} \quad \text{ja} \quad C_2 = \{-N(u), -R(w), -P(f(a), f(a)), -P(f(w), f(w))\}. \]

Klausuuleissa ei esiinny yhteisää muuttuja ja joukon

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

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

Esimerkki. (Faktorointi) Klausuulessa voi olla useita eri yhdistelmiä, klausuilenn \( \{P(x_1), P(y_1)\} \) ja \( \{-P(x_2), -P(y_2)\} \) yhdistelmiä ovat mm.

- \( \{P(x_1), -P(x_2)\} \) joukelle \( \{P(y_1), P(y_2)\} \) MGU \( \sigma = \{y_2/y_1\} \)

- tyhjä klausuli \( \square \) joukelle \( \{P(x_1), P(y_1), P(x_2), P(y_2)\} \) MGU

\[ \sigma = \{y_1/x_1, x_2/x_1, y_2/x_1\} \].

Esimerkki. Logiikahohdeinissa (PROLOG) laskenta askelet puhutuvat järjestetyn klausuulien väliseen resoluutioon.

Määritelmä. Olkoon \( G = \{-B_1(\bar{u}_1), \ldots, -B_m(\bar{u}_m)\} \) kyselyn negatiota vastaava järjestetty maaliklausuuli ja \( C = \{A(\bar{t}), -A_1(\bar{t}_1), \ldots, -A_n(\bar{t}_n)\} \) järjestetty ohjelmaklausuuli (ohjelman sääntö) siten, että

1. klausuellella \( G \) ja \( C \) ei ole yhteisää muuttuja ja
2. atomilla \( A(\bar{t}) \) ja valintafunction \( R \) määrittämässä literaalissa \( R(G) = -B_i(\bar{u}_i) \) esiintyvällä atomilla \( B_i(\bar{u}_i) \) on yleisin unifiointo \( \theta \).

Klausuuleihin \( G \) ja \( C \) yhdistelmäksi saadaan järjestetty maaliklausuuli

\[ G' = \{-B_1(\bar{u}_1), \ldots, -B_{i-1}(\bar{u}_{i-1}), -A_1(\bar{t}_1), \ldots, -A_{n-1}(\bar{t}_{n-1}), -B_{i+1}(\bar{u}_{i+1}), \ldots, -B_m(\bar{u}_m)\} \theta. \]

Esimerkki. Tyrppillinen PROLOGin erityispiirteitä

- Tyypillissä PROLOG toteutuksessa muuttujasymbolit erotetaan muista symboleista ison alkukirjaimen perusteella.
- Literaalijoukkoon siitä järjestetyt ohjelma- ja maaliklausulit kirjoitetaan säntöön seuraavalla tapaan:

\[ \{N(0)\} \sim n(0). \]

\[ \{N(s(x)), -N(x)\} \sim n(s(x)) \sim n(x). \]

\[ \{N(s(0)), -N(s(s(y)))\} \sim n(s(0)). \]

- Tyypillinen valintafunction valitsee maaliklausulin 1. atomin.

Esimerkki. Tyrppillinen PROLOG toteutuksessa seuraavat maaliklausulit:

1. \( n(0), n(s(s(Y))), \)
2. \( n(s(s(Y))), \)
3. \( n(s(Y)), \)
4. \( n(Y) \) ja
5. \( -tyhjä klausuuli. \)
8.2 Resoluutiloositukset

- Lähtökohtana on joukko klausuuleita $S$, jonka klausuuleista johdetaan uusia klausuuleita resoluutiosäännöllä.
- Johtojen ja hylykäksen määritelmät säilyvät ennallaan, mutta resoluutioskeleiden tulee täyttää resoluutiosäännön vaatimuksia.
- Tarvittaessa klausuulen muuttujat tulee nimetä uudelleen.
- Resoluutio on myös predikaattilogiikan tapauksessa virheetön ja täydellinen menettely klausuulijoukon toteutuvuuden tutkimiseen.

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

Todistus. Sivutetaan.

Muiden loogisten ongelmiin ratkominen

- Resoluutioilla voidaan selvittää lauseiden pätevyyttä ja logiista ekvivalenttia sekä tutkia lauseekon logiista seuraavuuska.
- Koska Skolemointi ei säilytä logiista ekvivalenttia vaan toteutuvuuden, nämä tulee muuntaa toteutuvuusongelmiksi.

Väite. Olkoon $\phi$ ja $\psi$ lauseita ja $\Sigma$ lausejoukko.
1. Pätevys: $\vdash \phi \iff \text{KM}(\{ \neg \phi \})$:lle löytyy hylykäs.
2. Ekvivalent: $\phi \equiv \psi \iff \text{KM}(\{ \phi \lor \psi \})$:lle löytyy hylykäs.
3. Logiinen seuraavus: $\Sigma \vdash \phi$
   $\iff$ klausuulijoukolle $\text{KM}(\Sigma \cup \{ \neg \phi \})$ löytyy hylykäs.

Yllä KM($\Gamma$) tarkoittaa lauseekon $\Gamma$ klausulimuotoa, mikä saadaan ottamalla yksittäisten lauseiden $\gamma \in \Gamma$ klausulimuotojen unioni.

Esimerkki. Osoitetaan predikaattilogiikan lausejoukko

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

toteutumattomaksi, haetaan lauseille ensin klausulikesytet:

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

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

$\implies S_1 \cup S_2$ on toteutumaton $\implies \Sigma$ on toteutumaton.
Esimerkki. Osoitetaan lause \( \exists x(E(x) \land K(x)) \) lausejoukon

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

loogiseksi seuraavaksi. Haetaan tarvittavat klausulimuodot:

\[
\forall x(I(x) \rightarrow E(x)) \quad \Rightarrow \quad \forall x(-I(x) \lor E(x))
\]

\[
\exists x(I(x) \land K(x)) \quad \Rightarrow \quad I(c) \land K(c)
\]

\[
-\exists x(E(x) \land K(x)) \quad \Rightarrow \quad \forall x(-E(x) \land -K(x))
\]

Kokonaisuutena saadaan sis klausulijoukko

\[
S = S_1 \cup S_2 \cup S_3 = \{-I(x), E(x), I(c), K(c), -E(x), -K(x)\}
\]

8.3 Ohjeita resoluutiotodistusten kirjoittamiseen

- Muuttujien uudelleennimeäminen on hyvä suorittaa systemaattisesti (esimerkiksi alaindekseen avulla).
- Yksittäistä klausulijoukon klausuluilla saattaa tarvita useita kertoja resoluutiotodistuksessa (jolloin muuttujien uudelleennimeäminen on välttämätöntä).
- Kirjoita yleisimmät unifioijat (MGU:t) näkyviin.
- Ellet kirjoita todistusta binääripuun muotoon, numeroi klausuluilla ja ilmoita, mistä klausuleista mikin klausuli on johdettu.
- Laske yleisimmien unifioijien kompositio selvitääksesi kyselyssä esiintyvillä muuttujilla arvot.
Esimerkki. Esitetään listat vakióen e (tyhjä lista) ja kaksipaikaisen funktiosymbolin c avulla (näin lista [1,2] saa esityksen c(1,c(2,e))).

Määriteltävät seuraavat listojen koskevat predikaatit:

1. \( K(x,e) \)
   - "listan x alkioina ovat listan y alkiot kääriteisessä järjestyksessä":
2. \( \forall x \forall y \forall z (K(x,y) \land L(y,v,z) \rightarrow K(c(x,v),z)) \)
3. \( L(y,v,z) = \text{"lista z on lista y, jonka perään on liitetty alkio v"}:
   - \( \forall x L(e,x,c(x,e)) \) ja
   - \( \forall x \forall y \forall z (L(y,v,z) \rightarrow L(c(x,y),v,c(x,z))) \).

Resoluutiotodistus:
1. \{ \neg K(c(1,c(2,e)),x_0) \} \quad P_5
2. \{ \neg K(x_1,y_1), \neg L(y_1,v_1,z_1), K(c(v_1,x_1),z_1) \} \quad P_2
3. \{ \neg K(c(2,e),y_1), \neg L(y_1,1,z_1) \} \quad 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) \} \quad P_2
5. \{ \neg K(e,y_2), \neg L(y_2,2,y_1), \neg L(y_1,1,z_1) \} \quad 3,4,MGU \{ v_2/2, x_2/e, z_2/y_1 \}
6. \{ K(e,e) \} \quad P_1
7. \{ \neg L(e,2,y_1), \neg L(y_1,1,z_1) \} \quad 5,6,MGU \{ y_2/e \}
8. \{ L(e,x_3,c(x_3,e)) \} \quad P_3
9. \{ \neg L(c(2,e),1,z_1) \} \quad 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)) \} \quad P_4
11. \{ \neg L(e,1,z_4) \} \quad 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)) \} \quad P_3
13. \square \quad 11,12,MGU \{ x_5/1, z_4/c(1,e) \}

- Unifiointien 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. vastaussubstituus).

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

Määrittelemä. Klaussulijoukon $S$ osajoukko $T$ on tukijoukko (engl. set of support), jos $S - T$ on toteutuva.

- **Tukijoukkkostrategiassa** ei miloinkaan suoriteta resoluutiotä joukon $S - T$ klaussuleille keskenään.

- **Tutkimasta logiista seuraavuutta** $\Sigma \models \phi$ lausejoukko $\Sigma$ (oleittamukset) on tyypillisesti toteutuva. Tällöin voidaan ajatella:

1. $T$ muodostuu lauseesta $\neg \phi$ saatavien klaussulien joukosta ja
2. joukkoon $S - T$ kuuluvat lausejoukosta $\Sigma$ saatavat klaussulit.

- Ristiriita aiheutuu siis konkreettisesti tukijoukon $T$ klaussuleista, mikäli $\Sigma \models \phi$ (eli $\Sigma \cup \{\neg \phi\}$ on toteutumaton).

Esimerkki. Tutkitaan seuraako lause $\phi \equiv \exists (E(y) \land K(y))$ loogisesti lausejoukosta $\Sigma = \{\forall x (I(x) \rightarrow E(x)), \exists x (I(x) \land K(x))\}$.

Lausejoukosta $\Sigma$ saadaan $S - T_0 = \{\{\neg I(x), E(x)\},\{I(c)\},\{K(c)\}\}$ ja lauseesta $\neg \phi$ tukijoukko $T_0 = \{\{\neg E(y), \neg K(y)\}\}$.

1. Valitaan klaussuli $C_1 = \{\neg E(y), \neg K(y)\} \in T_0$:
   - Klaussulista $\{K(c)\}$ saadaan $\{\neg E(c)\}$ (MGU $y/c$).
   - Klaussulista $\{\neg I(x), E(x)\}$ saadaan $\{\neg I(x), \neg K(x)\}$ (MGU $y/x$).

2. Valitaan $C_2 = \{\neg E(c)\} \in T_1 = \{\{\neg E(c)\},\{\neg I(x), \neg K(x)\}\}$:
   - Klaussulista $\{\neg I(x), E(x)\}$ saadaan $\{\neg I(x)\}$ (MGU $x/c$).

3. Valitaan $C_3 = \{\neg I(c)\} \in T_2 = \{\{\neg I(c)\},\{\neg I(x), \neg K(x)\}\}$:
   - Klaussulista $\{I(c)\}$ saadaan $\square$ (MGU $e$).

\[\rightarrow S \text{ ja } \Sigma \cup \{\neg \phi\} \text{ ovat toteutumattomia, joten } \Sigma \models \phi.\]
Motivaatio

Miksi tietokoneohjelmille tulisi kirjoittaa formaaleja espessiikäsiota?
- Spesifikointiota laadittaessa joudutaan suunnittelemaan ennalta varsin tarkaan mikä ohjelmiston on tarkoitus tehdä,
- Järjestelmän toteutus voidaan verifioida eli todeta määritellynsä mukaiseksi vasta, kun spesifikatio on tehty.
- Formaalisesta spesifiointissa etuna on määritelmien yksikäsitteisyys.
- Turvallisuusristiiset järjestelmat (esim. lentokoneen ohjaus- järjestelmät) vaativat perinpohjaista määritelyä ja verifiointia.
- Hyvin määritellyn ohjelman uudelleenkäyttöön on helpompaa.
Ohjelman tilojen esittäminen struktuurina

Määritelmä. Struktuuri $S$ on $Z$-struktuuri seuraavilla edellytyksillä:
(i) Struktuurin $S$ universumina $U$ on kokonaislukujen joukko $Z$.
(ii) Jokaisen kokonaisluvun (vakiosymboli tarkasteltavassa kielessä) tulkintana on kyseinen kokonaisluku itse.
(iii) Funktiosymbolien $+$ ja $*$ tulkintoja ovat yhteen-, vähennys- ja kertoaskufunktio kokonaislukujen joukossa.
(iv) Predikaattisymbolin $>$ tulkintana on suurempi kuin $-$relaatio kokonaislukujen joukossa.

Komentojen suorittamisen vaikutus tilaan

Määritelmä. Määritlemän tilansuoriteloaatio $S \rightarrow S'$ määrittelee tila $S'$, johon päädytään tilasta $S$, jos ja kun komennon $C$ suoritus päättyy.
1. Jos $C$ on sijoitusaause $x=E$, niin tila $S'$ on $S[x \mapsto E]$.
2. Jos $C$ on ketjulauseke $C_1 ; C_2$, niin $S'$ on tila, joka saavutetaan suorittamalla ensin $C_1$ ja sitten $C_2$.
3. Jos $C$ on ehtolauseke if $(B)$ then $(C_1)$ else $(C_2)$, 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 toistolauseke while $(B) \{ C_1 \}$ ja $S \not\models B$, niin $S' = S$.
5. Jos $C$ on toistolauseke while $(B) \{ C_1 \}$ ja $S \models B$, niin $S'$ on tila, joka saavutetaan suorittamalla $C_1$; while $(B) \{ C_1 \}$.

Huomioita.
- Ohjelman suorituksen *tila* voidaan rinnastaa $Z$-struktuurin $S$.
- Kokonaislukulausekkeen $E$ arvo tilassa $S$ on kokonaisluku $E^S$.
- Boolen 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 $(x \cdot x)$ ja $(z \cdot z)$ arvot ovat $(x \cdot x)^S = 4$ ja $(z \cdot z)^S = 9$.
Niinpä $S \models (x \cdot x < y) \& (y < z \cdot z)$, mutta $S \not\models (x \cdot y < z)$.

Esimerkki. *Tarkastellaan ohjelman*

$$y=1; z=1; \text{while}(y=x) \{ (z=z+1; y=y \cdot z) \}$$

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

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

muuttujan $y$ arvon on muuttujan $x$ arvon kertoama.

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

- Ohjelmointikielissä käytetään paljon ehtolausekkeita kontrolloimaan, millä ehdolla ja mitä toimintoja suoritetaan.
- Jos ehtolausekkeita muutetaan esim., optimointikäytänteissä, halutaan varmasti että toiminnat suoritetaan samoilla ehdolla.
- Ehtolausekkeiden ekvivalenssin osoittamiseen voidaan käyttää sekä lauseelogiikan että predikaattilogiikan menetelmiä.
- Jos ehtolausekkeiden evaluoinnilla on sivuaikutusen muutoksia ohjelman tilaan, pelkkä loogisen ekvivalenssin tarkastaminen ei välttämättä riitä.

**Esimerkki.** Tällainen sivuaikutus voi olla esim., virhetilanne, joka on aihetutun ehtojen evaluoinnista väärrässä järjestysessä.

**Esimerkki.** Vertaillaan kahta eri ohjelmaa:

```java
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., sijoituslauseke z=x suoritetaan niässä ohjelmissa seuraavilla ehdolla: \( A \land \neg C \land \neg (B \land C) \) ja \( A \land B \).
- Lauseelogiikan nojalla näiden välinen ekvivalenssi on looginen seurauksena \( \neg (B \land C) \), joka on aina voimassa B:n ja C:n välillä.

**Esimerkki.** (Jatkoa) ekvivalenssi voidaan todeta seuraavilla taululla:

\[
\begin{align*}
T \neg(B \land C) & \quad T \neg(B \land C) \\
T \neg(A \land B) & \quad T(A \land C) \land \neg(B \land C) \\
E(A \land C) \land \neg(B \land C) & \quad E(A \land B) \\
T A & \quad T A \\
T B & \quad T B \\
E A & \quad E B \\
& \quad \text{Näiden perusteella } \neg(B \land C) \vdash (A \land B) \leftrightarrow (A \land C) \land \neg(B \land C). \\
\end{align*}
\]

**Kytkenmät predikaattilogiikkaan**

- Boolen lauseke B on \( \mathbb{Z} \)-pätevä (merk., \( \models \mathbb{Z} B \)) \( \iff \) S \( \models B \) kaikissa \( \mathbb{Z} \)-struktseureissa S.
- Näin lausekkeiden muuttujat saavat universaalin tulkinnan.
- Boolen lausekkeet B₁ ja B₂ ovat \( \mathbb{Z} \)-ekvivalentit (merk., B₁ \( \equiv \mathbb{Z} B₂ \)) \( \iff \) lausekkeilla on sama totuusarvo kaikissa \( \mathbb{Z} \)-struktseureissa.
- Huomaa, että \( \models B \iff \models \mathbb{Z} B₁ ja B₁ \models B₂ \iff B₁ \equiv \mathbb{Z} B₂ \), mutta käntoiseita implikatit eivät välttämättä ole voimassa.

**Esimerkki.** \( \models \mathbb{Z} (((x>y) \land (y>x)), mutta \( \not\models \mathbb{Z} (((x>y) \land (y>x)), koska löytyy vastamalli S, jolle U = \{0\}, x^s = y^s = 0 ja \( >^S = \{0,0\} \).

\( \iff \) relation > suhde funktioihin +, - ja * joudutaan kuvaamaan erikseen (vrt., \( \neg(B \land C) \) edellä), jos käytetään predikaattilogiikkaa.
9.3 Ohjelmien esi- ja jälkiehdot

- Tarkasteltavan ohjelmointikohdan ohjelmalla on säätöön tilatavaruus, jonka läpäkäyminen on käytännössä mahdotonta.

- Joki mahdollisuus on tarkastella Boolean lausekkeiden $B$ määrittelemää tilaajoukkoa $\{S | S \models B\}$ ja analysoima, millaisia muutoksia annettu ohjelma näihin aiheuttaa.

- Mille tahansa ohjelmalle $P$ voidaan asettaa esi-ja jälkiehdot $B_1$ ja $B_2$ kirjoittamalla ns. Hoare-kolmikko $[B_1] \mathcal{P} [B_2]$.

- Karkeasti ottaen ajatuksena on, että esiehdon $B_1$ on tarkoitus taata jälkiehdon $B_2$ voimaantulo ohjelman $P$ suorituksen päättymättä.

**Esimerkki.** Olkoon Succ ohjelma $\text{if}(x==0)$ theen $y=1$ else $y=x+1$, jolle voidaan antaa spesiifikaatio: $\text{true}$ Succ $y=x+1$.

---

**Osittainen ja täysi oikeellisuus**

Olkoon $P$ ohjelma sekä $B_1$ ja $B_2$ kaksi Boolean lauseketta.

**Määritelmä.** Ohjelma $P$ on **osittais oikeellinen** annettujen esi-ja jälkiehdot $B_1$ ja $B_2$ suhteen (merk. $\models P \models B_1 \models B_2$) $\iff S \models B_2$ pätee saavutettavalle tilalle $S'$ aina kun ohjelman $P$ suoritus aloitetaan tilasta $S$, missä $S \models B_1$, ja ohjelman $P$ suoritus päätyy tilaan $S'$.

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

**Määritelmä.** Ohjelma $P$ on **täysin oikeellinen** annettujen esi-ja jälkiehdot $B_1$ ja $B_2$ suhteen (merk. $\models P \models B_1 \models B_2$) $\iff P \models B_1$ ja ohjelman $P$ suoritus päätyy aina kun $S \models B_1$ alkutilalle $S$.

**Huomio.** Vastaavat englannin kielisät termit ovat **partial correctness** ($\models P$) ja **total correctness** ($\models P$).
Heikoimat esitettävät

- Olkoon $P$ ohjelma $C_1; \ldots; C_n$, missä $C_1, \ldots, C_n$ ovat järjestyksessä peräkkäin suoritettavat komennot.

- Ominaisuudella $M = \{B_0, B_1, \ldots, B_n\}$ osoittaminen voidaan pilkkoosa ongelmille: tuli löytää sopiva ehdot $B_1, \ldots, B_{n-1}$ siten, että $M = \{B_0, B_1, \ldots, B_n\}$ on osoitettavissa kaikille $i \in \{1, \ldots, n\}$.

- Usein tällaiset ehdot voidaan löytää analysoimalla komentosekvenssien takaperin: haetaan komennolle $C_i$ (missä $i$ saa arvot $n, n-1, \ldots, 1$) haluttua esitettä $B_1$ siten, että $x = p[B_0, B_1, \ldots, B_n]$.

- Jatkossa tälläksiä todistusia kirjoitetaan sekenneisiä

Tarkastellaan seuraavaksi, millaisista todistusaskeleista tällainen sekennessi voidaan muodostaa edellä esittelyyn päättelysääntöjen nojalla.

1. Jos $B$ on jälkienno sijoituslausekkeella $x = E$, heikoimmaksi esiteltävä

   Voimme kirjata $B(x/E) \quad \vdash_{p} \quad B(x/E) \quad x = E \quad | \quad B$,

   **Esimerkki.** $x > 0 \quad y = x - 1 \quad y \geq 0$.

2. Jos $x = p[B_1, B_2]$ on jo osoitettu ja $B_0$ on ehdon $B_1$ vahehven

   $x = p[B_0, B_1, C_2, B_2]$ koska $x = p[B_0, C_2]$.

   **Esimerkki.** $x > 0 \quad y = x - 1 \quad y \geq 0$.

3. Jos $x = p[B_1, C_1, B_2]$ ja $x = p[B_2, C_2, B_3]$ ovat jo (rekursiivisesti)

   Osoitetut jälkienno lausekkeet $B_3$, niin lausekseen $\text{if}(B)$ siten $C_1$ else $C_2$

   Heikoimaksi esiteltävä kirjataan $(B \& B_1) \quad | \quad (!B \& B_2)$.

   **Esimerkki.** $x > y \quad | \quad (x > y) \quad | \quad (x > y) \quad | \quad (x > y)$

   **Esimerkki.** $x > y \quad | \quad (x > y) \quad | \quad (x > y) \quad | \quad (x = x)$

   **Esimerkki.** $(x > y) \quad | \quad (x > y)$

   if(x > y) then (x = x) else (x = y) = x = y

9.4 Toistolausekkeiden invariantti

- Ohehmoittamien keskeisiä primitiivejä ovat toistolausekkeet, joiden

   Avulla komentoja voitaisiin ohjelman haluttua määrrä.

   $z = 0; \quad v = 0; \quad \text{while}(z = x) \quad (z = z + 1; \quad v = v + y)$

- **Ongelma:** kuinka voitaisiin osoittaa toistorakenteita säätävien

   Algoritmin toimivuus kaikissa tilanteissa?

- Toistorakenteilla halutaan tyyppilemässä todistaa *invariantti* eli

   ominaisuus, joka säilyy voimassa toistorakenteen suorituksen ajan.

Määritelmä. Toistolausekeen **while**(B) C invarianttia I on mikä

Tehansa Boolen lauseke siten, että $p[B \& I] = C$.

Huomio. Invariantti I ei välttämättä ole jatkuvasti tosi komennon C

Suorituksen aikana, mutta ehdottomasti C:n suorituksen jälkeen.
9.5 Täysi oikeellisuus

- Tieto osittaisesta oikeellisuudesta (\( \models_p [B_1] C [B_2] \)) on hyödyllinen ainoastaan, mikäli komennon \( C \) suoritus todella päättyy.

- Täydellisen oikeellisuuden (\( \models \)) osoittamiseksi joudutaan todistamaan erikseen, että komennossa \( C \) esiintyvien toistolauseiden suoritus päättyy lopulta.

- Tätä varten tarvitaan vahvemmat päättelysääntöjä.

- Missä \( E \) on sopiva kokonaislukulauseke, \( n \) on (uusi) kokonaislukumuuttuja ja \( B_1 \) on vahvemmenttu invariantti \( B_\& (0 \leq E) \).

- Niin lausekkeen \( E \) arvo pienenee jatkuvasti toistettuessa \( C \):tä.


Esimerkki. Osoitetaan edellä annetun kertolaskuohjelman Multi osittainen oikeellisuus eli \( \models_p [true] Multi \{ v=x+y \}, \)

\[ [true] \{ 0 \leq 0 \} z=0 \{ 0=x+z \} \quad v=0 \{ v=x+y \} \quad (A4) \]

while(! (x==z)) {
    [v=x+y] \quad (A3)
    [v+y=(z+1)*y] \quad v=v+y \{ v=x+y \} \quad (A2)
}

[\{v=x+y\} \&\&(x==z)] \{ v=x+y \} \quad (A1)

Huomio. Todistuksessa käytetty invariantti on \( v=x+y \).

Edellä esitettyjä todistukselmaa (A1)...(A4) on merkitty ylös.

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