

## Sääntöpohjainen päättely

- Sääntöpohjainen päättely:  
logiikkaohjelmointi, deduktiviset tietokannat, asiantuntijajärjestelmät
- Horn-klausuulit tarjoavat sääntöpohjaiselle päättelylle formaalin semantiikan: voidaan määrittellä eksaktisti sääntöjoukosta saatavien johtopäätösten joukko.
- Horn-klausuuleille on kehitetty tehokkaita päättelymenetelmiä, mistä johtuen ne muodostavat laskennallisessa mielessä tärkeän lauselogiikan aliluokan.
- Tyypillisesti sovelluksissa tarvitaan kuitenkin Horn-klausuuleja ilmaisuvoimaisempi kieli (negaatio/komplementti).

## Ohjelmaklausuulit

Tyypillisesti käytetään ohjelmaklausuuleja (engl. program clauses; definite clauses).

**Määritelmä.** Ohjelmaklausuuli on literaalien disjunktio  $A \vee \neg B_1 \vee \dots \vee \neg B_n$ , jossa on täsmälleen *yksi* positiivinen literaali.

**Esimerkki.** Klausuuli  $\neg P \vee \neg Q \vee \neg R$  ei ole ohjelmaklausuuli, mutta  $\neg P \vee Q \vee \neg R$  (eli  $Q \leftarrow P \wedge R$ ) on.

**Väite.** Ohjelmaklausuulijoukko on toteutuva.

Ohjelmaklausuulijoukolle saadaan malli  $M$  esim. merkitsemällä jokaisen klausuulin  $A \vee \neg B_1 \vee \dots \vee \neg B_n$  positiivinen atomi  $A$  todeksi.

**Huomio.** Horn-klausuulien joukko ei ole välttämättä toteutuva. Esimerkiksi  $\{P, \neg P\}$  on toteutumaton.

## Horn-klausuulit

**Määritelmä.** Horn-klausuuli on literaalien disjunktio, jossa on korkeintaan *yksi* positiivinen literaali.

**Esimerkki.**  $\neg P \vee \neg Q \vee \neg R$  ja  $\neg P \vee Q \vee \neg R$  ovat Horn-klausuuleja.

**Esimerkki.** Klausuuli  $P \vee \neg Q \vee R$  ei ole Horn-klausuuli.

### Merkitätapa.

Horn-klausuuli  $\neg P_1 \vee \dots \vee \neg P_n \vee Q$  kirjoitetaan usein muodossa

$$Q \leftarrow P_1 \wedge \dots \wedge P_n$$

ja Horn-klausuuli  $\neg P_1 \vee \dots \vee \neg P_n$  muodossa

$$\leftarrow P_1 \wedge \dots \wedge P_n$$

➤ Merkitään jatkossa ohjelmaklausuulien joukossa  $S$  esiintyvien atomien joukkoa  $HB(S)$ :llä (kysymyksessä Herbrand-kanta).

➤ Horn-klausuulien joukolle  $S$  voidaan määrittellä vastaava ohjelmaklausuulijoukko  $S'$  ottamalla käyttöön uusi atomi  $\perp$  ja kirjoittamalla  $S$ :n puhtaasti negatiiviset Horn-klausuulit

$$\neg P_1 \vee \dots \vee \neg P_n$$

muotoon

$$\perp \vee \neg P_1 \vee \dots \vee \neg P_n \text{ (eli } \perp \leftarrow P_1 \wedge \dots \wedge P_n \text{)}.$$

☞ Horn-klausuulien toteutuvuusongelma voidaan ratkaista ohjelmaklausuuliesityksen avulla.

**Väite.** Joukko Horn-klausuuleja  $S$  on toteutuva  $\iff S' \not\models \perp$ , missä  $S'$  on joukkoa  $S$  vastaava ohjelmaklausuulien joukko.

**Esimerkki.**  $\{P, \neg P\}$  ei ole toteutuva ja  $\{P, \perp \leftarrow P\} \models \perp$ .

## Ohjelmaklausuulien minimimalli

**Määritelmä.** (i) Malli  $M_1$  (atomilauseiden joukko) on pienempi kuin malli  $M_2$  ( $M_1 < M_2$ ), joss  $M_1 \subset M_2$ .

(ii) Klausuulijoukon mallia  $M$  (atomilauseiden joukko) sanotaan minimaaliseksi, jos lausejoukolla ei ole sitä pienempää mallia.

**Esimerkki.** Tarkastellaan ohjelmaklausuulien joukkoa

$$S = \{Q \leftarrow R, R \leftarrow P \wedge Q\}.$$

- Totuusjaku  $M = \{Q, R\}$  on  $S$ :n malli.
- $M$  ei ole minimaalinen, koska  $M' = \emptyset$  on myös  $S$ :n malli.
- Sen sijaan  $M'$  on  $S$ :n minimimalli.

## Ordinaaliluvut

- Joukon  $S$  *transitiivisuus*: jokainen  $S$ :n alkio on  $S$ :n osajoukko.
- Joukko  $S$  on lineaarijärjestyksen  $<$  *hyvin järjestämä*  $\iff$  kaikille  $X \subseteq S$  löytyy pienin alkio  $x \in X$  järjestyksen  $<$  suhteen.
- Joukko  $S$  on ordinaali(luku), jos  $S$  on transitiviinen ja relaation  $\in$  hyvin järjestämä.
- Ordinaalien luokka on hyvin järjestetty:  $\alpha < \beta \iff \alpha \in \beta$ .
- Jos  $\alpha$  ja  $\beta$  ovat ordinaaleja, niin joko  $\alpha \subseteq \beta$  tai  $\beta \subseteq \alpha$ .

**Esimerkki.** Luonnolliset luvut vastaavat *äärellisiä* ordinaaleja:

$$0 \mapsto \emptyset, 1 = 0 + 1 \mapsto \{\emptyset\}, 2 = 1 + 1 \mapsto \{\emptyset, \{\emptyset\}\}, \dots$$

Luonnollisten lukujen joukko vastaa pienintä ääretöntä ordinaalia  $\omega$ .

**Teoreema.** Olkoon  $M_i \subseteq \text{HB}(S)$  (missä  $i \in I$ ) mikä tahansa kokoelma ohjelmaklausuulien joukon  $S$  malleja. Tällöin myös leikkaus

$$M = \bigcap \{M_i \mid i \in I\}$$

on malli joukolle  $S$ .

**Todistus.** Tehdään vasta oletus  $M \not\models S$ .

- $\implies \exists A \leftarrow B_1 \wedge \dots \wedge B_n \in S$  s.e.  $\{B_1, \dots, B_n\} \subseteq M$ ,  
mutta  $A \notin M$
- $\implies$  kaikille  $i \in I$  pätee  $\{B_1, \dots, B_n\} \subseteq M_i$
- $\implies$  kaikille  $i \in I$  pätee  $A \in M_i$ , koska  $M_i \models S$ ,  
 $A \leftarrow B_1 \wedge \dots \wedge B_n \in S$  ja  $M_i \models A \leftarrow B_1 \wedge \dots \wedge B_n$
- $\implies A \in M = \bigcap \{M_i \mid i \in I\}$ , ristiriita.  $\square$

## Ordinaaliluvut (jatkoa)

- Ordinaalin  $\alpha$  *seuraaja*  $\alpha + 1$  määritellään ordinaalina  $\alpha \cup \{\alpha\}$ .
- Jos  $\alpha = \beta + 1$ , ordinaali  $\alpha$  on *seuraajaordinaali*.
- *Rajaordinaali*  $\alpha$  on ordinaali, joka ei ole seuraajaordinaali.  
☞  $\emptyset$  ja  $\omega$  ovat ensimmäiset rajaordinaalit.
- Jokainen hyvinjärjestetty joukko on isomorfinen jonkin ordinaalin kanssa (tästä termi *ordinaali* eli järjestystyyppi, engl. *order type*).
- Ordinaalien  $\alpha$  ja  $\beta$  summalla  $\alpha + \beta$  tarkoitetaan näitä vastaavien järjestysten katenointia.  
☞  $2 + \omega = \omega$  ja  $\omega + 2$  eivät ole järjestyksinä isomorfis<sup>et</sup>.
- Ordinaali(luku)  $\alpha$  on *kardinaali(luku)*, jos  $|\alpha| \neq |\beta|$  kaikille  $\beta < \alpha$ .  
☞  $2$  ja  $\omega$  ovat kardinaaleja, mutta  $\omega + 2$  ei ( $|\omega| = |\omega + 2|$ ).

**Teoreema.** Ohjelmaklausuulijoukolla  $S$  on ainakin yksi minimimalli.

**Todistus.** Koska  $S$  on toteutuva, sillä on ainakin yksi malli  $M_0$ .

Määritellään laskeva mallien sekvenssi  $M_0, M_1, \dots$  seuraavasti:

1. Seuraajaordinaaleille  $\alpha$ :

- Jos  $M_{\alpha-1}$  on  $S$ :n minimimalli, määritellään  $M_\alpha = M_{\alpha-1}$
- Jos  $M_{\alpha-1}$  ei ole  $S$ :n minimimalli, on  $S$ :llä malli  $M \subset M_{\alpha-1}$ , jolloin määritellään  $M_\alpha = M$ .

2. Rajaordinaaleille  $\alpha$ :

- Määritellään  $M_\alpha = \bigcap_{\beta < \alpha} M_\beta$   
(joka on myös  $S$ :n mallien leikkauksena  $S$ :n malli).

Jos sekvenssi oletetaan aidosti laskevaksi päädytään ristiriitaan, kun  $|\alpha| > |\text{HB}(S)| \implies S$ :lle saadaan minimimalli jollakin ordinaalilla  $\alpha$ .

**Teoreema.** Joukolla ohjelmaklausuuleja  $S$  on yksikäsitteinen minimimalli  $M_S$  (pienin malli), joka on  $S$ :n kaikkien mallien leikkaus.

**Todistus.** Edellä osoitettiin, että  $S$ :llä on ainakin yksi minimimalli.

Oletetaan, että  $M_1$  ja  $M_2$  ovat  $S$ :n minimimalleja.

$$\begin{aligned} \implies M_1 \cap M_2 &\text{ on } S\text{:n malli.} \\ \implies M_1 \cap M_2 &= M_1 \text{ ja } M_1 \cap M_2 = M_2, \\ &\text{koska } M_1 \text{ ja } M_2 \text{ ovat minimimalleja} \\ \implies M_1 &= M_2. \end{aligned}$$

- Kaikille  $S$ :n malleille  $M \subseteq \text{HB}(S)$  pätee  $M_S \subseteq M$ , koska  $M_S$  osoitettiin edellä yksikäsitteiseksi.
- Täten kaikkien  $S$ :n mallien (joihin myös  $M_S$  lukeutuu) leikkaus  $\bigcap \{M \subseteq \text{HB}(S) \mid M \models S\}$  on täsmälleen  $M_S$ .  $\square$

**Esimerkki.** Tarkastellaan ääretöntä ohjelmaklausuulien joukkoa

$$S = \{P_0 \leftarrow P_0, P_1 \leftarrow P_1, P_2 \leftarrow P_2, \dots\}.$$

- Totuusjaku  $M_0 = \text{HB}(S) = \{P_0, P_1, P_2, \dots\}$  on joukon  $S$  malli, muttei minimaalinen, koska myös  $M_1 = \{P_1, P_2, P_3, \dots\} \models S$ .
- Yleistäen: totuusjaku  $M_i = \{P_i, P_{i+1}, P_{i+2}, \dots\}$  on joukon  $S$  malli, muttei minimaalinen, koska  $M_{i+1} = \{P_{i+1}, P_{i+2}, P_{i+3}, \dots\}$  on myös joukon  $S$  malli.
- Rajaordinaalilla  $\omega$  saadaan joukolle  $S$  minimimalli

$$M_\omega = \bigcap_{i < \omega} M_i = \emptyset.$$

- Huomaa, että  $|\omega| = |\text{HB}(S)|$ .

**Teoreema.** Olkoon  $S$  joukko ohjelmaklausuuleja. Tällöin

$$M_S = \{P \in \text{HB}(S) \mid S \models P\}.$$

**Todistus.** Tarkastellaan mitä hyvänsä atomia  $P \in \text{HB}(P)$ . Nyt

$$\begin{aligned} S \models P &\iff M \models P \text{ kaikille } S\text{:n malleille } M \subseteq \text{HB}(S) \\ &\iff P \in M \text{ kaikille } S\text{:n malleille } M \subseteq \text{HB}(S) \\ &\iff P \in M_S, \end{aligned}$$

koska  $M_S = \bigcap \{M \subseteq \text{HB}(S) \mid M \models S\}$ .

## Minimimallin konstruointi

**Määritelmä.** Olkoon  $S$  (mahd. ääretön) joukko ohjelmaklausuuleja. Määritellään operaattori  $T_S : 2^{\text{HB}(S)} \rightarrow 2^{\text{HB}(S)}$  seuraavasti:

$$T_S(M) = \{P \mid P \leftarrow P_1 \wedge \dots \wedge P_n \in S \text{ ja } \{P_1, \dots, P_n\} \subseteq M\}$$

**Esimerkki.** Olkoon  $S = \{P \leftarrow P, Q, R_1 \leftarrow Q, R_2 \leftarrow Q \wedge P\}$ .

Nyt  $T_S(\{P\}) = \{P, Q\}$  ja  $T_S(\{P, Q\}) = \{P, Q, R_1, R_2\}$ .

- $M$  on kuvauksen  $T_S$  kiintopiste, joss  $T_S(M) = M$ .

**Esimerkki.**  $M_1 = \{P, Q, R_1, R_2\}$  on kuvauksen  $T_S$  kiintopiste:

$$T_S(M_1) = \{P, Q, R_1, R_2\} = M_1$$

- Kiintopiste  $M$  on pienin, jos muille kiintopisteille  $M', M \subseteq M'$ .

**Esimerkki.**  $M_2 = \{Q, R_1\}$  on kuvauksen  $T_S$  pienin kiintopiste.

## Määritelmä.

$$T_S \uparrow 0 = \emptyset$$

$$T_S \uparrow \alpha = T_S(T_S \uparrow \alpha - 1)$$

$$T_S \uparrow \omega = \bigcup \{T_S \uparrow \alpha \mid \alpha < \omega\}$$

**Teoreema.** Pienin kiintopiste  $\text{lfp}(T_S) = T_S \uparrow \omega$ .

**Esimerkki.** Joukolle  $S = \{P \leftarrow P, Q, R_1 \leftarrow Q, R_2 \leftarrow Q \wedge P\}$ :

$$T_S \uparrow 0 = \emptyset,$$

$$T_S \uparrow 1 = T_S(T_S \uparrow 0) = T_S(\emptyset) = \{Q\},$$

$$T_S \uparrow 2 = T_S(T_S \uparrow 1) = T_S(\{Q\}) = \{Q, R_1\},$$

$$T_S \uparrow 3 = T_S(T_S \uparrow 2) = T_S(\{Q, R_1\}) = \{Q, R_1\},$$

⋮

$$T_S \uparrow \omega = \bigcup \{T_S \uparrow \alpha \mid \alpha < \omega\} = \{Q, R_1\}.$$

- Operaattori  $T_S$  on monotoninen:

$$\text{kaikille } M \subseteq M' \subseteq \text{HB}(S) \text{ pätee } T_S(M) \subseteq T_S(M').$$

- Knaster-Tarski -teoreema: jokaisella monotonisella operaattorilla on yksikäsitteinen pienin (suurin) kiintopiste.

**Teoreema.** Operaattorilla  $T_S$  on pienin kiintopiste

$$\text{lfp}(T_S) = \bigcap \{M \subseteq \text{HB}(S) \mid T_S(M) \subseteq M\}.$$

**Väite.** Atomilauseiden joukko  $M \subseteq \text{HB}(S)$  on ohjelmaklausuulijoukon  $S$  malli, joss  $T_S(M) \subseteq M$ .

**Todistus.**  $M \not\models S$

$$\iff \exists A \leftarrow B_1 \wedge \dots \wedge B_n \in S \text{ s.e. } \{B_1, \dots, B_n\} \subseteq M, \text{ mutta } A \notin M$$

$$\iff \exists A \in T_S(M) \text{ s.e. } A \notin M \iff T_S(M) \not\subseteq M. \quad \square$$

**Teoreema.** Pienin malli  $M_S = \text{lfp}(T_S)$ .

## Johdettavuus ohjelmalauseilla

**Määritelmä.** Atomilauseen  $P$  on johto ohjelmalauseista  $S$  on äärellinen jono atomilauseita  $P_0, \dots, P_n$  siten, että  $P = P_n$  ja

(i)  $P_0 \in S$  ja

(ii) jokaiselle  $i > 0$  on olemassa  $P_i \leftarrow Q_1 \wedge \dots \wedge Q_n \in S$  siten, että  $\{Q_1, \dots, Q_n\} \subseteq \{P_1, \dots, P_{i-1}\}$ .

Jos lauseelle  $P$  on olemassa johto joukosta  $S$ , merkitään tätä  $S \vdash P$ .

**Teoreema.** (i)  $S \models P \iff S \vdash P$  ja

(ii)  $M_S = \{P \mid P \text{ on atomilause ja } S \vdash P\}$ .

**Esimerkki.**  $S = \{P \leftarrow P, Q, R_1 \leftarrow Q, R_2 \leftarrow Q \wedge R_1, R_2 \leftarrow P\}$ .

Jono  $Q, R_1, R_2$  on lauseen  $R_2$  johto joukosta  $S$  ( $S \vdash R_2$ ).

Jono  $Q, P, R_2$  ei ole lauseen  $R_2$  johto joukosta  $S$ .

## Logiikkaohjelmat ja deduktiiviset tietokannat

- Logiikkaohjelmat (ilman ei-logisia operaatioita: cut, not, assert, retract) ja deduktiiviset tietokannat voidaan tulkita ohjelmaklausuulien joukoiksi  $S$ , joissa literaalit sisältävät muuttujia sekä vakio- ja funktiosymboleja.

- Säännöt ovat siis muotoa

$$P(\vec{t}) \leftarrow P_1(\vec{t}_1) \wedge \dots \wedge P_n(\vec{t}_n)$$

missä  $\vec{t}, \vec{t}_1, \dots, \vec{t}_n$  ovat predikaattien  $P, P_1, \dots, P_n$  argumentteina olevia termilistoja.

- Mikä on tällaisen ohjelmaklausuulijoukon  $S$  semantiikka: milloin annettuun kyselyyn  $Q(\vec{t})$  pitää antaa myönteinen vastaus?

## Esimerkki. Deduktiivinen tietokanta $S$ :

$p(a,c). \quad p(b,c). \quad q(X) :- p(X,Y).$

- Herbrand-universumi  $HU(S) = \{a, b, c\}$ .
- Herbrand-instanssien joukko  $S_H$ :

$p(a,c). \quad p(b,c).$

$q(a) :- p(a,a). \quad q(a) :- p(a,b). \quad q(a) :- p(a,c).$

$q(b) :- p(b,a). \quad q(b) :- p(b,b). \quad q(b) :- p(b,c).$

$q(c) :- p(c,a). \quad q(c) :- p(c,b). \quad q(c) :- p(c,c).$

- Pienin Herbrand-malli  $M_{S_H} = \{p(a,c), p(b,c), q(a), q(b)\}$ .
- $M_{S_H}$  antaa kyselyille odotetut vastaukset:

?-  $p(a,c)$ . kyllä                      ?-  $p(a,d)$ . ei

?-  $q(a)$ . kyllä                         ?-  $q(c)$ . ei

- Jokaiseen ohjelmaklausuulien joukkoon  $S$  liittyy Herbrand-universumi  $HU(S)$  eli  $S$ :n vakio- ja funktiosymboleista muodostettavissa olevien muuttujattomien termien joukko.

- Mikä tahansa  $S$ :n muuttujia sisältävä ohjelmaklausuuli  $P(\vec{t}) \leftarrow P_1(\vec{t}_1) \wedge \dots \wedge P_n(\vec{t}_n)$  voidaan *instantioida* korvaamalla muuttujat  $HU(S)$ : muuttujattomien termien eri kombinaatioilla.

- Muuttujia sisältävä ohjelmaklausuulien joukko  $S$  edustaa *Herbrand-instanssiensa* joukkoa  $S_H$ , joka voidaan tulkita propositionaaliseksi ohjelmaklausuulien joukoksi.

- Joukolla  $S_H$  on yksikäsitteinen pienin Herbrand malli  $M_{S_H}$ .

- Tästä saadaan kriteeri kyselyjen oikeellisuudelle: olkoon  $x_1, \dots, x_n$  kyselyssä  $Q(\vec{t})$  esiintyvät muuttujat.

Nyt  $S \models \exists x_1 \dots \exists x_n Q(\vec{t}) \iff$  on olemassa vastaussubstituutio  $\theta$  (korvaa  $x_i$ :t  $HU(S)$ :n termeillä) s.e.  $Q(\vec{t})\theta \in M_{S_H}$ .

## Esimerkki. Olkoon tietokanta $S$ seuraava:

$\text{vanhempi}(\text{kaija}, \text{jussi}). \quad \text{vanhempi}(\text{tauno}, \text{jussi}).$

$\text{vanhempi}(\text{kaija}, \text{teiija}). \quad \text{vanhempi}(\text{saija}, \text{kaija}).$

$\text{vanhempi}(\text{raiija}, \text{saija}).$

$\text{sisarus}(X, Y) :- \text{vanhempi}(Z, X), \text{vanhempi}(Z, Y).$

$\text{esivanhempi}(X, Y) :- \text{vanhempi}(X, Y).$

$\text{esivanhempi}(X, Y) :- \text{vanhempi}(X, Z), \text{esivanhempi}(Z, Y).$

- Pienin Herbrand-malli  $M_{S_H}$  antaa odotetut vastaukset:

?-  $\text{vanhempi}(\text{kaija}, \text{jussi})$ . kyllä                      ?-  $\text{vanhempi}(\text{raiija}, \text{jussi})$ . ei

?-  $\text{sisarus}(\text{teiija}, \text{jussi})$ . kyllä                      ?-  $\text{sisarus}(\text{teiija}, \text{jukka})$ . ei

?-  $\text{esivanhempi}(\text{raiija}, \text{jussi})$ . kyllä                      ?-  $\text{esivanhempi}(\text{raiija}, \text{jari})$ . ei

## Sääntökielen ilmaisuvoima

Säännöillä pystytään ilmaisemaan keskeiset relaatioalgebran operaatiot:

1. Unioni:  $P(x_1, \dots, x_n) \leftarrow Q_1(x_1, \dots, x_n)$   
 $P(x_1, \dots, x_n) \leftarrow Q_2(x_1, \dots, x_n)$
2. Leikkaus:  $P(x_1, \dots, x_n) \leftarrow Q_1(x_1, \dots, x_n) \wedge Q_2(x_1, \dots, x_n)$
3. Projektio:  $\text{Vanhempi}(x) \leftarrow \text{Vanhempi}(x, y)$
4. Valinta:  $\text{Miljonaari}(x) \leftarrow \text{Tulot}(x, y) \wedge (y > 1000000)$
5. Kompositio:  $\text{Tenttitulos}(x, y) \leftarrow \text{Opiskelija}(x, i) \wedge \text{Arvosana}(i, y)$

- Verrattuna relaatioalgebraan (SQL): komplementti puuttuu.
- Toisaalta säännöt ovat ilmaisuvoimaisempia kuin relaatioalgebra (SQL): relaatioiden määritelmät voivat olla rekursiivisia.

**Esimerkki.** Transitivinen sulkeuma:

$$\text{Yhteys}(x, y) \leftarrow \text{Lento}(x, y)$$

$$\text{Yhteys}(x, y) \leftarrow \text{Lento}(x, y) \wedge \text{Yhteys}(y, z)$$

- Tyypillisesti lisätään komplementti jossain muodossa.

**Esimerkki.** Ad hoc -ratkaisut johtavat ongelmiin:

$$\text{Oluenjuoja}(x) \leftarrow \text{DI}(x) \wedge \text{not Absolutisti}(x),$$

$$\text{Absolutisti}(x) \leftarrow \text{Ekonomi}(x) \wedge \text{not Oluenjuoja}(x),$$

$$\text{DI}(\text{Liisa}), \text{Ekonomi}(\text{Liisa})$$