Sääntöpohjainen päättely

- Sääntöpohjainen päättely:
  logiikkaohjelmointi, deduktiiviset tietokannat, asiantuntijajärjestelmät
- Horn-klausuulit tarjoavat sääntöpohjaiselle päättelylle formaalin semantiikan: voidaan määritellä 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).

Hornklausuulit

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.

Merkintätapa.
Horn-klausuuli $\neg P_1 \vee \cdots \vee \neg P_n \vee Q$ kirjoitetaan usein muodossa

$$Q \leftarrow P_1 \wedge \cdots \wedge P_n$$

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

$$\leftarrow P_1 \wedge \cdots \wedge P_n$$

Ohjelmaklausuulit

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

Määritelmä. *Ohjelmaklausuuli* on literaalien disjunktio $A \vee \neg B_1 \vee \cdots \vee \neg B_n$, 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. Jokainen ohjelmaklausuulijoukko on toteutuva.

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

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

- Merkitään jatkossa ohjelmaklausuulien joukossa S esiintyvien atomien joukkoa $\mathrm{HB}(S)$:llä (kysymyksessä Herbrand-kanta).
- Horn-klausuulien joukolle S voidaan määritellä vastaava ohjelmaklausuulijoukko S' ottamalla käyttöön uusi atomi $\perp$ ja kirjoittamalla S:n puhtaasti negatiiviset Horn-klausuulit

  $$\neg P_1 \vee \cdots \vee \neg P_n$$

  muotoon

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

☞ 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) Totuusjakelu M_1 (atomilauseiden joukko) on pienempi kuin totuusjakelu 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

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

– Totuusjakelu $M = \{Q, R\}$ on S:n malli.

– M ei ole minimaalinen, koska $M' = \emptyset$ on myös S:n malli.

– Sen sijaan M' on S:n minimimalli.

Teoreema. Olkoon $M_i \subseteq \mathrm{HB}(S)$ (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 vastaoletus $M \not\models S$.

$\Longrightarrow$ $\exists A \leftarrow B_1 \wedge \cdots \wedge B_n \in S$ s.e. $\{B_1, \ldots, B_n\} \subseteq M$, mutta $A \notin M$

$\Longrightarrow$ kaikille $i \in I$ pätee $\{B_1, \ldots, B_n\} \subseteq M_i$

$\Longrightarrow$ kaikille $i \in I$ pätee $A \in M_i$, koska $M_i \models S$, $A \leftarrow B_1 \wedge \cdots \wedge B_n \in S$ ja $M_i \models A \leftarrow B_1 \wedge \cdots \wedge B_n$

$\Longrightarrow$ $A \in M = \bigcap \{M_i \mid i \in I\}$, ristiriita. $\square$

Ordnaaliluvut

- 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 α ja β ovat ordinaaleja, niin joko $\alpha \subseteq \beta$ tai $\beta \subseteq \alpha$.

Esimerkki. Luonnolliset luvut vastaavat *äärellisiä* ordinaaleja:

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

Luonnollisten lukujen joukko vastaa pienintä ääretöntä ordinaalia ω.

Ordnaaliluvut (jatkoa)

- Ordinaalin α *seuraaja* $\alpha + 1$ määritellään ordinaalina $\alpha \cup \{\alpha\}$.
- Jos $\alpha = \beta + 1$, ordinaali α on *seuraajaordinaali*.
- *Rajaordinaali* α on ordinaali, joka ei ole seuraajaordinaali.
  ☞ $\emptyset$ ja ω ovat ensimmäiset rajaordinaalit.
- Jokainen hyvinjärjestetty joukko on isomorfinen jonkin ordinaalin kanssa (tästä termi *ordinaali* eli järjestystyyppi, engl. *order type*).
- Ordinaalien α ja β summalla $\alpha + \beta$ tarkoitetaan näitä vastaavien järjestysten katenointia.
  ☞ $2 + \omega = \omega$ ja $\omega + 2$ eivät ole järjestyksinä isomorfiset.
- Ordinaali(luku) α on *kardinaali(luku)*, jos $|\alpha| \neq |\beta|$ kaikille $\beta < \alpha$.
  ☞ 2 ja ω 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 $M_0, M_1, \ldots$ seuraavasti:

1. Seuraajaordinaaleille α:
   - 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 α:
   - 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 $|\alpha| > |\mathrm{HB}(S)| \implies S$:lle saadaan minimimalli jollakin ordinaalilla α.

Esimerkki. Tarkastellaan ääretöntä ohjelmaklausuulien joukkoa

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

- Totuusjakelu $M_0 = \mathrm{HB}(S) = \{P_0, P_1, P_2, \ldots\}$ on joukon S malli, muttei minimaalinen, koska myös $M_1 = \{P_1, P_2, P_3, \ldots\} \models S$.
- Yleistäen: totuusjakelu $M_i = \{P_i, P_{i+1}, P_{i+2}, \ldots\}$ on joukon S malli, muttei minimaalinen, koska $M_{i+1} = \{P_{i+1}, P_{i+2}, P_{i+3}, \ldots\}$ on myös joukon S malli.
- Rajaordinaalilla ω saadaan joukolle S minimimalli
$$M_\omega = \bigcap_{i<\omega} M_i = \emptyset.$$
- Huomaa, että $|\omega| = |\mathrm{HB}(S)|$.

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.

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

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

Teoreema. Olkoon S joukko ohjelmaklausuuleja. Tällöin

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

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

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

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

Minimimallin konstruointi

Määritelmä. Olkoon S (mahd. ääretön) joukko ohjelmaklausuuleja. Määritellään operaattori $\mathrm{T}_S : 2^{\mathrm{HB}(S)} \to 2^{\mathrm{HB}(S)}$ seuraavasti:

$$\mathrm{T}_S(M) = \{P \mid P \leftarrow P_1 \wedge \cdots \wedge P_n \in S \text{ ja } \{P_1, \ldots, P_n\} \subseteq M\}$$

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

Nyt $\mathrm{T}_S(\{P\}) = \{P, Q\}$ ja $\mathrm{T}_S(\{P, Q\}) = \{P, Q, R_1, R_2\}$.

- M on kuvauksen T_S *kiintopiste*, joss $\mathrm{T}_S(M) = M$.
  Esimerkki. $M_1 = \{P, Q, R_1, R_2\}$ on kuvauksen T_S kiintopiste: $\mathrm{T}_S(M_1) = \{P, Q, R_1, R_2\} = M_1$
- Kiintopiste M on pienin, jos muille kiintopisteille M', $M \subseteq M'$.
  Esimerkki. $M_2 = \{Q, R_1\}$ on kuvauksen T_S pienin kiintopiste.

- Operaattori T_S on monotoninen:

$$\text{kaikille } M \subseteq M' \subseteq \mathrm{HB}(S) \text{ pätee } \mathrm{T}_S(M) \subseteq \mathrm{T}_S(M').$$

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

Teoreema. Operaattorilla T_S on pienin kiintopiste

$$\mathrm{lfp}(\mathrm{T}_S) = \bigcap \{M \subseteq \mathrm{HB}(S) \mid \mathrm{T}_S(M) \subseteq M\}.$$

Väite. Atomilauseiden joukko $M \subseteq \mathrm{HB}(S)$ on ohjelmaklausuulijoukon S malli, joss $\mathrm{T}_S(M) \subseteq M$.

Todistus. $M \not\models S$
$\iff \exists A \leftarrow B_1 \wedge \cdots \wedge B_n \in S$ s.e. $\{B_1, \ldots, B_n\} \subseteq M$, mutta $A \notin M$
$\iff \exists A \in \mathrm{T}_S(M)$ s.e. $A \notin M \iff \mathrm{T}_S(M) \not\subseteq M$. $\square$

Teoreema. Pienin malli $M_S = \mathrm{lfp}(\mathrm{T}_S)$.

Määritelmä.
$\mathrm{T}_S \uparrow 0 = \emptyset$
$\mathrm{T}_S \uparrow \alpha = \mathrm{T}_S(\mathrm{T}_S \uparrow \alpha - 1)$
$\mathrm{T}_S \uparrow \omega = \bigcup \{\mathrm{T}_S \uparrow \alpha \mid \alpha < \omega\}$

Teoreema. Pienin kiintopiste $\mathrm{lfp}(\mathrm{T}_S) = \mathrm{T}_S \uparrow \omega$.

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

$$\begin{aligned}
&\mathrm{T}_S \uparrow 0 = \emptyset,\\
&\mathrm{T}_S \uparrow 1 = \mathrm{T}_S(\mathrm{T}_S \uparrow 0) = \mathrm{T}_S(\emptyset) = \{Q\},\\
&\mathrm{T}_S \uparrow 2 = \mathrm{T}_S(\mathrm{T}_S \uparrow 1) = \mathrm{T}_S(\{Q\}) = \{Q, R_1\},\\
&\mathrm{T}_S \uparrow 3 = \mathrm{T}_S(\mathrm{T}_S \uparrow 2) = \mathrm{T}_S(\{Q, R_1\}) = \{Q, R_1\},\\
&\vdots\\
&\mathrm{T}_S \uparrow \omega = \bigcup \{\mathrm{T}_S \uparrow \alpha \mid \alpha < \omega\} = \{Q, R_1\}.
\end{aligned}$$

Johdettavuus ohjelmalauseilla

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

(i) $P_0 \in S$ ja

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

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

Teoreema. (i) $S \models P \iff S \vdash P$ ja
(ii) $M_S = \{P \mid P$ 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-loogisia 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

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

missä $\vec{t}$, $\vec{t}_1$, ..., $\vec{t}_n$ ovat predikaattien P, P_1, ..., P_n argumentteina olevia termilistoja.
- Mikä on tällaisen ohjelmaklausuulijoukon S semantiikka: milloin annettuun kyselyyn $Q(\vec{t})$ pitää antaa myönteinen vastaus?

- Jokaiseen ohjelmaklausuulien joukkoon S liittyy Herbrand-universumi $\mathrm{HU}(S)$ eli S:n vakio- ja funktiosymboleista muodostettavissa olevien muuttujattomien termien joukko.
- Mikä tahansa S:n muuttujia sisältävä ohjelmaklausuuli $P(\vec{t}) \leftarrow P_1(\vec{t}_1) \wedge \cdots \wedge P_n(\vec{t}_n)$ voidaan *instantioida* korvaamalla muuttujat $\mathrm{HU}(S)$: 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 $M_{S_{\mathrm{H}}}$.
- Tästä saadaan kriteeri kyselyjen oikeellisuudelle: olkoon $x_1, \ldots, x_n$ kyselyssä $Q(\vec{t})$ esiintyvät muuttujat.

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

Esimerkki. Deduktiivinen tietokanta S:

```
p(a,c).  p(b,c).  q(X) :- p(X,Y).
```

- Herbrand-unversumi $\mathrm{HU}(S) = \{\mathtt{a},\mathtt{b},\mathtt{c}\}$.
- Herbrand-instanssien joukko S_{H}:

```
p(a,c). p(b,c).
q(a) :- p(a,a).  q(a) :- p(a,b).  q(a) :- p(a,c).
q(b) :- p(b,a).  q(b) :- p(b,b).  q(b) :- p(b,c).
q(c) :- p(c,a).  q(c) :- p(c,b).  q(c) :- p(c,c).
```

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

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


Esimerkki. Olkoon tietokanta S seuraava:

```
vanhempi(kaija,jussi). vanhempi(tauno,jussi).
vanhempi(kaija,teija). vanhempi(saija,kaija).
vanhempi(raija,saija).
sisarus(X,Y) :- vanhempi(Z,X), vanhempi(Z,Y).
esivanhempi(X,Y) :- vanhempi(X,Y).
esivanhempi(X,Y) :- vanhempi(X,Z), esivanhempi(Z,Y).
```

- Pienin Herbrand-malli $M_{S_{\mathrm{H}}}$ antaa odotetut vastaukset:

```
?- vanhempi(kaija,jussi). kyllä      ?- vanhempi(raija,jussi). ei
?- sisarus(teija,jussi). kyllä       ?- sisarus(teija,jukka). ei
?- esivanhempi(raija,jussi). kyllä   ?- esivanhempi(raija,jari). ei
```

Sääntökielen ilmaisuvoima

Säännöillä pystytään ilmaisemaan keskeiset relaatioalgebran operaatiot:

1. Unioni: $P(x_1,\ldots,x_n) \leftarrow Q_1(x_1,\ldots,x_n)$
   $P(x_1,\ldots,x_n) \leftarrow Q_2(x_1,\ldots,x_n)$
2. Leikkaus: $P(x_1,\ldots,x_n) \leftarrow Q_1(x_1,\ldots,x_n) \wedge Q_2(x_1,\ldots,x_n)$
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 rekursivisia.

  Esimerkki. Transitivinen sulkeuma:

  $\text{Yhteys}(x,y) \leftarrow \text{Lento}(x,y)$
  $\text{Yhteys}(x,y) \leftarrow \text{Lento}(x,z) \wedge \text{Yhteys}(z,y)$
- Tyypillisesti lisätään komplementti jossain muodossa.

  Esimerkki. Ad hoc -ratkaisut johtavat ongelmiin:

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