Sääntöpohjainen päättely (jatkoa)

- Horn-klausuulit tarjoavat loogisen pohjan sääntöpohjaiselle päättelylle.
- Tyypillisesti tarvitaan kuitenkin Horn-klausuuleja ilmaisuvoimaisempi kieli, jossa negaatio/komplementti: logiikkaohjelmointi, deduktiiviset tietokannat, asiantuntijajärjestelmät
- Rekursion ja negaation yhdistäminen ei onnistu suoraviivaisesti.
  ☞ Stabiilien mallien semantiikka
- Toteutustekniikat edenneet merkittävästi viime vuosina.

Negaation lisääminen Horn-klausuuleihin

- Relaation komplementti tarvitaan useimmissa sovellutuksissa.

$$\mathrm{Absolutisti}(X) \leftarrow \mathrm{Ekonomi}(X) \wedge \mathrm{not}\ \mathrm{Oluenjuoja}(X)$$

- Ad hoc -ratkaisut johtavat ongelmiin: sääntöjen semantiikka tulee helposti riippuvaiseksi toteutusyksityiskohdista, kuten esim.
  – sääntöjen talletusjärjestyksestä, tai
  – sääntötulkin noudattamasta päättelyalgoritmista.

Esimerkki. Tarkastellaan seuraavaa sääntöjoukkoa:

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

Mitä tulisi vastata kyselyyn $\mathrm{Oluenjuoja}(\mathrm{Liisa})$?

Stabiilien mallien semantiikka

- M. Gelfond and V. Lifschitz [1988]: *The Stable Model Semantics for Logic Programming*.
- Käsitellään *normaaleja logiikkaohjelmia*, joiden säännöt ovat muotoa $H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \mathrm{not}\ B_{m+1} \wedge \cdots \wedge \mathrm{not}\ B_n$.
- Stabiilien mallien idea:
  (i) Ehto not P on tosi mallissa $M \iff P \notin M$, ja
  (ii) Malli M on stabiili, jos se on pienin Herbrand-malli niille säännöille, joiden kaikki not-ehdot ovat tosia mallissa M.

Esimerkki. Tarkastellaan edellisessä esimerkissä annetun normaalin logiikkaohjelman Herbrand-instanssia (vakion Liisa suhteen):

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

Malli $M = \{\mathrm{DI}(\mathrm{Liisa}), \mathrm{Ekonomi}(\mathrm{Liisa}), \mathrm{Oluenjuoja}(\mathrm{Liisa})\}$ on stabiili:

- sääntöjen $\mathrm{Oluenjuoja}(\mathrm{Liisa}) \leftarrow \mathrm{DI}(\mathrm{Liisa})$, $\mathrm{DI}(\mathrm{Liisa})$ ja $\mathrm{Ekonomi}(\mathrm{Liisa})$ not-ehdot toteutuvat, ja
- M on näiden sääntöjen pienin Herbrand-malli.

Huomio. Symmetriasyistä myös malli

$$M' = \{\mathrm{DI}(\mathrm{Liisa}), \mathrm{Ekonomi}(\mathrm{Liisa}), \mathrm{Absolutisti}(\mathrm{Liisa})\}$$

on stabiili.

Määritelmä. Olkoon P muuttujaton ohjelma ja $M \subseteq \mathrm{HB}(P)$.

Ohjelman P Gelfond-Lifschitz-*redukti* P^M on sääntöjoukko, joka saadaan poistamalla ohjelmasta P

(i) jokainen sääntö, jossa on ehto $\mathrm{not}\, B$ siten, että $B \in M$ ja

(ii) kaikki not-ehdot jäljelle jäävistä säännöistä.

Määritelmä. Olkoon P muuttujaton logiikkaohjelma. Muuttujattomien atomilauseiden joukko M on ohjelman P stabiili malli $\iff$ M on ohjelmaklausuulijoukon P^M pienin malli, eli

$$M = \mathrm{lfp}(\mathrm{T}_{P^M}).$$

☞ Stabiilit mallit ovat kiintopisteyhtälön ratkaisuja.

Esimerkki. Tarkastellaan logiikkaohjelmaa $P =$

$$\begin{aligned}\{\, & A \leftarrow C \wedge \mathrm{not}\, B, \\ & B \leftarrow \mathrm{not}\, A, \\ & C \leftarrow \mathrm{not}\, D, \\ & D \leftarrow \mathrm{not}\, A \,\}.\end{aligned}$$

Nyt esim.

- $M_1 = \{A, C\}$ on P:n stabiili malli, koska $P^{M_1} = \{A \leftarrow C, C\}$ ja M_1 on P^{M_1}:n pienin malli.
- $M_2 = \{A, D\}$ ei ole P:n stabiili malli, koska $P^{M_2} = \{A \leftarrow C\}$ ja tämän pienin malli on $\{\}$.
- $M_3 = \{B, D\}$ on myös P:n stabiili malli.

- $P^M = \{\, H \leftarrow B_1 \wedge \cdots \wedge B_m \mid$
  $H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \mathrm{not}\, B_{m+1} \wedge \cdots \wedge \mathrm{not}\, B_n \in P,$
  $\{B_{m+1}, \ldots, B_n\} \cap M = \emptyset \,\}$
- Määritellään operaattori

  $$\Gamma_P(M) = \{A \mid A \in \mathrm{HB}(P) \text{ ja } P^M \models A\} = \mathrm{lfp}(\mathrm{T}_{P^M}).$$

- Näin M on ohjelman P stabiili malli $\iff$ $M = \Gamma_P(M)$.
- Γ_P on antimonotoninen: jos $M \subseteq M'$, niin $\Gamma_P(M') \subseteq \Gamma_P(M)$, koska jos $M \subseteq M'$, niin $P^{M'} \subseteq P^M$.
- Stabiilit mallit ovat *minimaalisia*: jos $M \subseteq \mathrm{HB}(P)$ on ohjelman P stabiili malli, mikään $M' \subset M$ ei ole ohjelman P stabiili malli.
- Stabiilit mallit ovat *hyvin perustuvia*: jos M on ohjelman P stabiili malli, niin $A \in M \iff P^M \models A$.

Muuttujalliset ohjelmat

- Logiikkaohjelmalle P määritellään Herbrand-instanssien joukko P_{H} kuten Horn-klausuulien joukoille.

Määritelmä. Olkoon P mahdollisesti muuttujia sisältävä logiikkaohjelma ja P_{H} ohjelman P Herbrand-instanssi.

Muuttujattomien atomilauseiden joukko $M \subseteq \mathrm{HB}(P)$ on ohjelman P stabiili malli $\iff$ M on P_{H}^M:n pienin malli (eli $M = \Gamma_{P_{\mathrm{H}}}(M)$).

Esimerkki. Olkoon $P = \{\, A(1,2),\ B(x) \leftarrow A(x,y) \wedge \mathrm{not}\, B(y) \,\}$.

$P_{\mathrm{H}} = \{\, B(1) \leftarrow A(1,1) \wedge \mathrm{not}\, B(1),\ B(1) \leftarrow A(1,2) \wedge \mathrm{not}\, B(2),$
$A(1,2),\ B(2) \leftarrow A(2,1) \wedge \mathrm{not}\, B(1),\ B(2) \leftarrow A(2,2) \wedge \mathrm{not}\, B(2) \,\}.$

- Nyt esim. $M = \{A(1,2),\ B(1)\}$ on P:n stabiili malli.

Esimerkkejä

Logiikkaohjelmat ja stabiilit mallit tarjoavat varsin joustavan tavan mallittaa myös dynaamisia tilanteita.

- Sääntö, jolla poikkeuksia:
  $\mathrm{Flies}(x) \leftarrow \mathrm{Bird}(x) \wedge \mathrm{not}\ \mathrm{Abnormal}(x)$,
  $\mathrm{Abnormal}(x) \leftarrow \mathrm{Penguin}(x)$,
  $\mathrm{Abnormal}(x) \leftarrow \mathrm{Wingless}(x)$,
  $\mathrm{Abnormal}(x) \leftarrow \mathrm{Oily}(x)$, jne.
- Sääntö, jolla vaihtoehtoisia johtopäätöksiä:
  $\mathrm{Lo}(x) \leftarrow \mathrm{not}\ \mathrm{Hi}(x) \wedge \mathrm{Nondet}(x) \wedge \mathrm{Precond}(x)$ ja
  $\mathrm{Hi}(x) \leftarrow \mathrm{not}\ \mathrm{Lo}(x) \wedge \mathrm{Nondet}(x) \wedge \mathrm{Precond}(x)$.
- Eheysehtojen ja vaatimusten esittäminen:
  $F \leftarrow \mathrm{Nogood} \wedge \mathrm{not}\ F$ ja $F \leftarrow \mathrm{not}\ \mathrm{Necessary} \wedge \mathrm{not}\ F$.

Rajoiteohjelmointi säännöillä

- Tyypillisen logiikkaohjelmointijärjestelmän tehtävänä on vastata kyselyyn "kyllä" (ja antaa vastaussubstituutio θ) tai "ei".
- Stabiilit mallit tarjoavat mahdollisuuden logiikkaohjelman *rajoiteohjelmointitulkintaan:*
  (i) Ohjelman säännöt nähdään rajoitteina ohjelman malleille.
  (ii) Ratkaisuina ovat atomien joukot (eli stabiilit mallit).
  (iii) Rajoiteohjelmointijärjestelmän tehtävänä on hakea annetulle logiikkaohjelmalle ratkaisujoukkoja.
- Stabiileihin malleihin perustuva rajoiteohjelmointi on varsin ilmaisuvoimaista:
  Monet rajoiteohjelmointitehtävät voidaan suoraviivaisesti palauttaa ongelmaksi löytää logiikkaohjelmalle stabiili malli.

Esimerkki: lauselogiikan toteutuvuusongelma

Muunnetaan klausuulijoukko S logiikkaohjelmaksi P_S seuraavasti:

1. Jokaiselle $A \in \mathrm{HB}(S)$ otetaan käyttöön uusi atomi $\hat{A}$ ja kaksi sääntöä $\hat{A} \leftarrow \mathrm{not}\ A$ ja $A \leftarrow \mathrm{not}\ \hat{A}$.
2. Jokaista S:n klausuulia $A_1 \vee \cdots \vee A_n \vee \neg B_1 \vee \cdots \vee \neg B_m$ kohti otetaan käyttöön sääntö
   $$F \leftarrow \hat{A}_1 \wedge \cdots \wedge \hat{A}_n \wedge B_1 \wedge \cdots \wedge B_m \wedge \mathrm{not}\ F$$
   missä F on uusi atomi.

Väite. Klausuulijoukola S on malli M (eli S on toteutuva) $\Longleftrightarrow$ ohjelmalla P_S on stabiili malli M' siten, että $M = M' \cap \mathrm{HB}(S)$.

Esimerkki. Olkoon $S = \{A \vee B,\ A \vee \neg B,\ \neg A \vee \neg B\}$.

Logiikkaohjelmaesitykseksi saadaan $P_S =$

$$\{\, A \leftarrow \mathrm{not}\ \hat{A},\ \hat{A} \leftarrow \mathrm{not}\ A,\ B \leftarrow \mathrm{not}\ \hat{B},\ \hat{B} \leftarrow \mathrm{not}\ B,$$
$$F \leftarrow \hat{A} \wedge \hat{B} \wedge \mathrm{not}\ F,\ F \leftarrow \hat{A} \wedge B \wedge \mathrm{not}\ F,\ F \leftarrow A \wedge B \wedge \mathrm{not}\ F\,\}.$$

- Nyt klausuulijoukola S on malli M $\Longleftrightarrow$ ohjelmalla P_S on stabiili malli M' siten, että $M = M' \cap \{A, B\}$.
- Koska $M_1' = \{A, \hat{B}\}$ on P_S:n stabiili malli, tiedämme, että $M_1 = \{A\}$ on S:n malli.
- Toisaalta esim. $M_2' = \{\hat{A}, \hat{B}\}$ ei ole P_S:n stabiili malli.

Esimerkki: graafin 3-väritys

- Esitetään kytketty graafi G tietokantana käyttäen predikaattia $\mathrm{Kaari}(x,y)$ = "solmujen x ja y välillä on kaari".
- Muodostetaan P_G^{3c} lisäämällä tietokantaan seuraavat säännöt:

$$\mathrm{Solmu}(x) \leftarrow \mathrm{Kaari}(x,y),\ \mathrm{Solmu}(y) \leftarrow \mathrm{Kaari}(x,y),$$
$$\mathrm{Musta}(x) \leftarrow \mathrm{Solmu}(x) \wedge \mathrm{not}\ \mathrm{Valkoinen}(x) \wedge \mathrm{not}\ \mathrm{Harmaa}(x),$$
$$\mathrm{Valkoinen}(x) \leftarrow \mathrm{Solmu}(x) \wedge \mathrm{not}\ \mathrm{Musta}(x) \wedge \mathrm{not}\ \mathrm{Harmaa}(x),$$
$$\mathrm{Harmaa}(x) \leftarrow \mathrm{Solmu}(x) \wedge \mathrm{not}\ \mathrm{Valkoinen}(x) \wedge \mathrm{not}\ \mathrm{Musta}(x),$$
$$F \leftarrow \mathrm{Kaari}(x,y) \wedge \mathrm{Musta}(x) \wedge \mathrm{Musta}(y) \wedge \mathrm{not}\ F,$$
$$F \leftarrow \mathrm{Kaari}(x,y) \wedge \mathrm{Valkoinen}(x) \wedge \mathrm{Valkoinen}(y) \wedge \mathrm{not}\ F \text{ ja}$$
$$F \leftarrow \mathrm{Kaari}(x,y) \wedge \mathrm{Harmaa}(x) \wedge \mathrm{Harmaa}(y) \wedge \mathrm{not}\ F.$$

Väite. Graafilla G on 3-väritys $\iff$ ohjelmala P_G^{3c} on stabiili malli.

Esimerkki: Hamiltonin silmukka

- Esitetään kytketty graafi G tietokantana kuten edellä.
- Muodostetaan P_G^{H} lisäämällä tietokantaan säännöt seuraavasti.

1. Erotetaan solmut kaarien perusteella:

$$\mathrm{Solmu}(x) \leftarrow \mathrm{Kaari}(x,y) \text{ ja } \mathrm{Solmu}(y) \leftarrow \mathrm{Kaari}(x,y).$$

2. Valitaan silmukalle alkusolmu:

$$\mathrm{Alkusolmu}(x) \leftarrow \mathrm{Solmu}(x) \wedge \mathrm{not}\ \mathrm{Muusolmu}(x),$$
$$\mathrm{Muusolmu}(x) \leftarrow \mathrm{Solmu}(x) \wedge \mathrm{not}\ \mathrm{Alkusolmu}(x),$$
$$F \leftarrow \mathrm{Alkusolmu}(x) \wedge \mathrm{Alkusolmu}(y) \wedge \mathrm{Solmu}(x) \wedge \mathrm{Solmu}(y) \wedge$$
$$\mathrm{not}\ (x = y) \wedge \mathrm{not}\ F,$$
$$\mathrm{Onalkusolmu} \leftarrow \mathrm{Solmu}(x) \wedge \mathrm{Alkusolmu}(x) \text{ ja}$$
$$F \leftarrow \mathrm{not}\ \mathrm{Onalkusolmu} \wedge \mathrm{not}\ F.$$

3. Valitaan silmukkaan kuuluvat kaaret:

$$\mathrm{Silmukassa}(x1,x2) \leftarrow \mathrm{Kaari}(x1,x2) \wedge \mathrm{not}\ \mathrm{Ulkona}(x1,x2),$$
$$\mathrm{Ulkona}(x1,x2) \leftarrow$$
$$\mathrm{Kaari}(x1,x2) \wedge \mathrm{Silmukassa}(x1,x3) \wedge \mathrm{not}\ (x2 = x3) \text{ ja}$$
$$\mathrm{Ulkona}(x1,x2) \leftarrow$$
$$\mathrm{Kaari}(x1,x2) \wedge \mathrm{Silmukassa}(x3,x2) \wedge \mathrm{not}\ (x1 = x3).$$

4. Vaaditaan, että kaikki graafin solmut ovat saavutettavissa:

$$\mathrm{Saavutettavissa}(x) \leftarrow \mathrm{Alkusolmu}(x),$$
$$\mathrm{Saavutettavissa}(x) \leftarrow$$
$$\mathrm{Kaari}(y,x) \wedge \mathrm{Silmukassa}(y,x) \wedge \mathrm{Saavutettavissa}(y) \text{ ja}$$
$$F \leftarrow \mathrm{Solmu}(x), \mathrm{not}\ \mathrm{Saavutettavissa}(x), \mathrm{not}\ F.$$

Väite. Ohjelmalla P_G^{H} on stabiili malli $\iff$ G:llä on Hamiltonin silmukka (kaikkien solmujen kautta täsmälleen kerran kulkeva alkusolmuun palaava polku).

Kompakti karakterisointi

- not-ehdoilla on keskeinen rooli stabiilien mallien määrittämisessä.
- $\mathrm{NA}(P) \subseteq \mathrm{HB}(P)$: ohjelmassa P not-ehdoissa esiintyvät atomilauseet (engl. *negative antecedents*).
- Stabiilit mallit voidaan karakterisoida joukon $\mathrm{NA}(P)$ osajoukoilla.
- Määritellään joukolle not-iteraaleja F: *deduktiivinen sulkeuma* $\mathrm{DCl}(P,F)$ on pienin malli ohjelmaklausuulien joukolle

$$P(F) = \{\ H \leftarrow B_1 \wedge \cdots \wedge B_m \mid$$
$$H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \mathrm{not}\ B_{m+1} \wedge \cdots \wedge \mathrm{not}\ B_n \in P,$$
$$\{\mathrm{not}\ B_{m+1}, \ldots, \mathrm{not}\ B_n\} \subseteq F\ \}.$$

Huomio. Sulkeuma $\mathrm{DCl}(P,F) = \Gamma_P(\{A \in \mathrm{HB}(P) \mid \mathrm{not}\ A \notin F\})$.

Määritelmä. Joukko F not-literaaleja on P-*täysi*, joss kaikille $A \in \mathrm{NA}(P)$ pätee $\mathrm{not}\, A \in F \iff A \notin \mathrm{DCl}(P,F)$.

Teoreema. Olkoon P muuttujaton ohjelma.

(i) Jos F on P-täysi, $\mathrm{DCl}(P,F)$ on P:n stabiili malli.

(ii) Jos M on P:n stabiili malli, $F = \{\mathrm{not}\, A \mid A \in \mathrm{NA}(P) - M\}$ (eli M:n komplementti joukon $\mathrm{NA}(P)$ suhteen) on P-täysi ja $M = \mathrm{DCl}(P,F)$.

Esimerkki. Joukko $\mathrm{NA}(P) = \{A,B,D\}$ logiikkaohjelmalle $P = \{A \leftarrow C \wedge \mathrm{not}\, B,\ B \leftarrow \mathrm{not}\, A,\ C \leftarrow \mathrm{not}\, D,\ D \leftarrow \mathrm{not}\, A\}$.

- not-literaalien joukko $F_1 = \{\mathrm{not}\, B, \mathrm{not}\, D\}$ on P-täysi, koska $P(F_1) = \{A \leftarrow C, C\}$ ja $\mathrm{DCl}(P,F_1) = \{A,C\}$.
- Mutta esim. $F_2 = \{\mathrm{not}\, B\}$ ei ole P-täysi, koska $P(F_2) = \{A \leftarrow C\}$ ja sulkeuma $\mathrm{DCl}(P,F_2) = \{\}$.

Stabiilien mallien approksimointi

- A. Van Gelder, K.A. Ross, J.S. Schlipf [1988]: *The Well-founded Semantics for General Logic Programs.*
- P:n stabiileja malleja voidaan approksimoida operaattorilla Γ_P.
- Olkoon M joukko "varmoja" johtopäätöksiä (aluksi esim. $\emptyset$).

(i) $P^M = \{\, H \leftarrow B_1 \wedge \cdots \wedge B_m \mid H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \mathrm{not}\, B_{m+1} \wedge \cdots \wedge \mathrm{not}\, B_n \in P,\ \{B_{m+1}, \ldots, B_n\} \cap M = \emptyset \,\}$

(ii) Joukko $\Gamma_P(M)$ antaa "potentiaaliset" johtopäätökset.

(iii) Joukko $\Gamma_P^2(M) = \Gamma_P(\Gamma_P(M))$ antaa "varmat johtopäätökset".

- Operaattori Γ_P^2 on monotoninen, koska Γ_P on antimonotoninen.
- Operaattorilla on pienin kiintopiste $\mathrm{lfp}(\Gamma_P^2)$ (Knaster-Tarski).

Väite. Kaikille ohjelman P stabiileille malleille M pätee: $\mathrm{lfp}(\Gamma_P^2) \subseteq M$ ja $M \subseteq \Gamma_P(\mathrm{lfp}(\Gamma_P^2))$.

☞ Operaattori Γ_P antaa P:n stabiileille malleille ala- ja ylärajan.

Määritelmä. Muuttujattoman ohjelman P WF-malli on literaalijoukko

$$\mathrm{WFM}(P) = \mathrm{lfp}(\Gamma_P^2) \cup \{\mathrm{not}\, A \mid A \in \mathrm{HB}(P) \text{ ja } A \notin \Gamma_P(\mathrm{lfp}(\Gamma_P^2))\}.$$

- Kysymyksessä *osittaismalli* (kolmiarvoinen malli), joka voidaan täydentää ohjelman P *totaalimalliksi* (eli kaksiarvoiseksi malliksi), joita ovat mm. stabiilit mallit.
- WF-malli on aina olemassa (mutta $\mathrm{lfp}(\Gamma_P^2)$ saavuttamiseksi saatetaan tarvita yli ω iteraatiota, jos P on ääretön).

Esimerkki. Olkoon $P = \{A \leftarrow \mathrm{not}\, A\}$.

Tällöin $\Gamma_P(\emptyset) = \{A\}$ ja $\Gamma_P^2(\emptyset) = \emptyset$, joten $\mathrm{WFM}(P) = \emptyset$.

Esimerkki. Tarkastellan ohjelmaa $P' =$

$$\{\, A_1 \leftarrow \mathrm{not}\, A_0,\quad A_2 \leftarrow \mathrm{not}\, A_1,\quad A_3 \leftarrow \mathrm{not}\, A_2,\quad B_1 \leftarrow A_3 \wedge \mathrm{not}\, B_2,\quad B_2 \leftarrow A_3 \wedge \mathrm{not}\, B_1 \,\}.$$

1. Ensin $\Gamma_{P'}(\emptyset) = \{A_1,A_2,A_3,B_1,B_2\}$ ja $\Gamma_{P'}^2(\emptyset) = \{A_1\}$.
2. Edelleen $\Gamma_{P'}(\{A_1\}) = \{A_1,A_3,B_1,B_2\}$ ja $\Gamma_{P'}^2(\{A_1\}) = \{A_1,A_3\}$.
3. Lopulta $\Gamma_{P'}(\{A_1,A_3\}) = \{A_1,A_3,B_1,B_2\}$ ja $\Gamma_{P'}^2(\{A_1,A_3\}) = \{A_1,A_3\}$.
4. Siis $\mathrm{WFM}(P') = \{A_1,A_3,\mathrm{not}\, A_0,\mathrm{not}\, A_2\}$.