

Epämonotoninen sääntöpohjainen päättely

Käsiteltäviä asioita:

1. Tarve negatiivisille ehdoille
2. Stabiilien mallien semantiikka
3. Muuttujalliset ohjelmat
4. Sääntöpohjainen rajoiteohjelmointi
5. Kompakti karakterisointi
6. Stabiilien mallien approksimointi

Negaatiosta aiheutuvia ongelmia

- Relaaation komplementti tarvitaan useimmissa sovelluksissa.


$$\text{Absolutisti}(X) \leftarrow \text{Ekonomi}(X) \wedge \text{not Oluenjuoja}(X)$$
- Ad hoc -ratkaisut johtavat ongelmiin: sääntöjen semantiikka tulee helposti riippuvaiseksi toteutusyksityiskohdista, kuten esim.
 - sääntöjen talletusjärjestyksestä, tai
 - sääntötulkin noudattamasta päättelyalgoritmista.

Esimerkki. Tarkastellaan seuraavaa sääntöjoukkoa:

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

Mitä tulisi vastata kyselyyn Oluenjuoja(Liisa)?

1. Tarve negatiivisille ehdoille

- Horn-klausuulien minimimalleihin perustuva semantiikka tarjoaa loogisen pohjan sääntöpohjaiselle päättelylle.
- Tyypillisesti tarvitaan kuitenkin Horn-klausuuleja ilmaisuvoimaisempi kieli, jossa negatio/komplementti: logiikkaohjelmointi, deduktiiviset tietokannat, asiantuntijajärjestelmät
- Rekursion ja negaation yhdistäminen ei onnistu suoraviivaisesti.
 Stabiilien mallien semantiikka
- Toteutustekniikat edenneet merkittävästi viimeisten kymmenen vuoden aikana.

2. Stabiilien mallien semantiikka

- M. Gelfond and V. Lifschitz [1988]: *The Stable Model Semantics for Logic Programming*.
- Käsitellään *normaaleja logiikkaohjelmia*, joiden säännöt ovat muotoa $H \leftarrow B_1 \wedge \dots \wedge B_m \wedge \text{not } B_{m+1} \wedge \dots \wedge \text{not } B_n$.
- Stabiilien mallien idea:
 - (i) Ehto $\text{not } P$ 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):

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

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

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

Huomio. Symmetriasyistä myös malli

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

on stabiili.

Esimerkki. Tarkastellaan logiikkaohjelmaa $P =$

$$\{ A \leftarrow C \wedge \text{not } B, \\ B \leftarrow \text{not } A, \\ C \leftarrow \text{not } D, \\ D \leftarrow \text{not } A \}.$$

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.



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

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

- (i) jokainen sääntö, jossa on ehto $\text{not } B$ 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

$$M = \text{lfp}(T_{P^M}).$$

 Stabiilit mallit ovat kiintopisteyhtälön ratkaisuja.

Γ_P -operaattori

- $P^M = \{ H \leftarrow B_1 \wedge \dots \wedge B_m \mid \\ H \leftarrow B_1 \wedge \dots \wedge B_m \wedge \text{not } B_{m+1} \wedge \dots \wedge \text{not } B_n \in P, \\ \{B_{m+1}, \dots, B_n\} \cap M = \emptyset \}$

- Määritellään operaattori

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

- Näin M on ohjelman P stabiili malli $\iff M = \Gamma_P(M)$.
- Γ_P on *antimonotoninen* (järjestyksen vaihtava):
Jos $M \subseteq M'$, niin $P^{M'} \subseteq P^M$ ja edelleen $\Gamma_P(M') \subseteq \Gamma_P(M)$.

Stabiilien mallien ominaisuuksia

- Stabiilit mallit ovat *minimaalisia*: jos $M \subseteq \text{HB}(P)$ on ohjelman P stabiili malli, mikään $M' \subset M$ ei ole ohjelman P stabiili malli.

Todistus. Oletetaan, että ohjelmalla P on stabiilit mallit M ja M' siten että $M' \subseteq M$.

$$\implies M = \Gamma_P(M) \subseteq \Gamma_P(M') = M'$$

$$\implies M = M'.$$

Täten oletus kahden stabiilin mallin välisestä suhteesta $M' \subset M$ (eli $M' \subseteq M$ ja $M' \neq M$) johtaa ristiriitaan.

- Stabiilit mallit ovat *hyvin perustuvia*: jos M on ohjelman P stabiili malli, niin $A \in M \iff P^M \models A$.

Esimerkkejä

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

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

3. 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 $M \subseteq \text{HB}(P)$ on ohjelman P stabiili malli $\iff M$ on P_H^M :n pienin malli (eli $M = \Gamma_{P_H}(M)$).

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

$P_H = \{ B(1) \leftarrow A(1,1) \wedge \text{not } B(1), B(1) \leftarrow A(1,2) \wedge \text{not } B(2), A(1,2), B(2) \leftarrow A(2,1) \wedge \text{not } B(1), B(2) \leftarrow A(2,2) \wedge \text{not } B(2) \}$.

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

4. Sääntöpohjainen rajoiteohjelmointi

- Tyypillisen logiikkaohjelmointijärjestelmän tehtävänä on vastata kyselyyn "kyllä" (ja antaa vastaussubstituutio θ) tai "ei".
- Stabiilit mallit tarjoavat mahdollisuuden logiikkaohjelman *rajoiteohjelmointitulkin*taan:
 - (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 $A \in \text{HB}(S)$ otetaan käyttöön uusi atomi \hat{A} ja kaksi sääntöä $\hat{A} \leftarrow \text{not } A$ ja $A \leftarrow \text{not } \hat{A}$.
2. Jokaista S :n klausuulia $A_1 \vee \dots \vee A_n \vee \neg B_1 \vee \dots \vee \neg B_m$ kohti otetaan käyttöön sääntö

$$F \leftarrow \hat{A}_1 \wedge \dots \wedge \hat{A}_n \wedge B_1 \wedge \dots \wedge B_m \wedge \text{not } F$$

missä F on uusi atomi.

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

Esimerkki: graafin 3-väritys

➤ Esitetään kytketty graafi G tietokantana käyttäen predikaattia $\text{Kaari}(x, y) = \text{"solmujen } x \text{ ja } y \text{ välillä on kaari"}$.

➤ Muodostetaan P_G^{3c} lisäämällä tietokantaan seuraavat säännöt:

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

$$\text{Musta}(x) \leftarrow \text{Solmu}(x) \wedge \text{not Valkoinen}(x) \wedge \text{not Harmaa}(x),$$

$$\text{Valkoinen}(x) \leftarrow \text{Solmu}(x) \wedge \text{not Musta}(x) \wedge \text{not Harmaa}(x),$$

$$\text{Harmaa}(x) \leftarrow \text{Solmu}(x) \wedge \text{not Valkoinen}(x) \wedge \text{not Musta}(x),$$

$$F \leftarrow \text{Kaari}(x, y) \wedge \text{Musta}(x) \wedge \text{Musta}(y) \wedge \text{not } F,$$

$$F \leftarrow \text{Kaari}(x, y) \wedge \text{Valkoinen}(x) \wedge \text{Valkoinen}(y) \wedge \text{not } F \text{ ja}$$

$$F \leftarrow \text{Kaari}(x, y) \wedge \text{Harmaa}(x) \wedge \text{Harmaa}(y) \wedge \text{not } F.$$

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

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

Logiikkaohjelmaesitykseksi saadaan $P_S =$

$$\{A \leftarrow \text{not } \hat{A}, \hat{A} \leftarrow \text{not } A, B \leftarrow \text{not } \hat{B}, \hat{B} \leftarrow \text{not } B,$$

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

- Nyt klausuulijoukolla S on malli $M \iff$ 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: Hamiltonin silmukka

➤ Esitetään kytketty graafi G tietokantana kuten edellä.

➤ Muodostetaan P_G^H lisäämällä tietokantaan säännöt seuraavasti.

1. Erotetaan solmut kaarien perusteella:

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

2. Valitaan silmukalle alkusolmu:

$$\text{Alkusolmu}(x) \leftarrow \text{Solmu}(x) \wedge \text{not Muusolmu}(x),$$

$$\text{Muusolmu}(x) \leftarrow \text{Solmu}(x) \wedge \text{not Alkusolmu}(x),$$

$$F \leftarrow \text{Alkusolmu}(x) \wedge \text{Alkusolmu}(y) \wedge \text{Solmu}(x) \wedge \text{Solmu}(y) \wedge$$

$$\text{not } (x = y) \wedge \text{not } F,$$

$$\text{Onalkusolmu} \leftarrow \text{Solmu}(x) \wedge \text{Alkusolmu}(x) \text{ ja}$$

$$F \leftarrow \text{not Onalkusolmu} \wedge \text{not } F.$$

3. Valitaan silmukkaan kuuluvat kaaret:

$$\text{Silmukassa}(x1,x2) \leftarrow \text{Kaari}(x1,x2) \wedge \text{not Ulkona}(x1,x2),$$

$$\text{Ulkona}(x1,x2) \leftarrow$$

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

$$\text{Ulkona}(x1,x2) \leftarrow$$

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

4. Vaaditaan, että kaikki graafin solmut ovat saavutettavissa:

$$\text{Saavutettavissa}(x) \leftarrow \text{Alkusolmu}(x),$$

$$\text{Saavutettavissa}(x) \leftarrow$$

$$\text{Kaari}(y,x) \wedge \text{Silmukassa}(y,x) \wedge \text{Saavutettavissa}(y) \text{ ja}$$

$$F \leftarrow \text{Solmu}(x), \text{not Saavutettavissa}(x), \text{not } F.$$

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

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

Teoreema. Olkoon P muuttujaton ohjelma ja F joukko not-literaaleja.

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

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

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

5. Kompakti karakterisointi

- not-ehdoilla on keskeinen rooli stabiilien mallien määrittämisessä.
- Määritellään $\text{NA}(P) \subseteq \text{HB}(P)$ (engl. *negative antecedents*) ohjelman P not-ehdoissa esiintyvien atomilauseiden joukkona.
- Stabiilit mallit voidaan karakterisoida joukon $\text{NA}(P)$ osajoukoilla.
- Määritellään not-iteraalien joukolle F *deduktiivinen sulkeuma* $\text{DCI}(P,F)$, joka on pienin malli ohjelmaklausuulien joukolle

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

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

6. Stabiilien mallien approksimointi

- A. Van Gelder, K.A. Ross, J.S. Schlipf [1988]: *The Well-founded Semantics for General Logic Programs*.
- Ohjelman P stabiileja malleja voidaan approksimoida operaattorin Γ_P avulla seuraavasti.
- Olkoon M joukko "varmoja" johtopäätöksiä (aluksi esim. \emptyset).
 - (i) $P^M = \{ H \leftarrow B_1 \wedge \dots \wedge B_m \mid \\ H \leftarrow B_1 \wedge \dots \wedge B_m \wedge \text{not } B_{m+1} \wedge \dots \wedge \text{not } B_n \in P, \\ \{B_{m+1}, \dots, B_n\} \cap M = \emptyset \}$
 - (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 $\text{lfp}(\Gamma_P^2)$ (Knaster-Tarski).

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

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

Määritelmä. Muuttujattoman ohjelman P WF-malli on literaalijoukko

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

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

Oppimistavoitteet

- Tiedät mitä ongelmia negaation ja rekursion yhdistämisestä seuraa.
- Ymmärrät syvällisesti stabiilien mallien määritelmän.
- Osaat hakea yksinkertaiselle logiikkaohjelmalle stabiilit mallit ja well-founded-mallin.
- Pystyt ratkaisemaan yksinkertaisia rajoiteohjelmointiongelmia kuvaamalla näiden ratkaisut sääntöjen avulla.

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

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

Esimerkki. Tarkastellaan ohjelmaa $P' =$

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

1. Ensinnä $\Gamma_{P'}(\emptyset) = \{A_1, A_2, A_3, B_1, B_2\}$ 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 $\text{WFM}(P') = \{A_1, A_3, \text{not } A_0, \text{not } A_2\}$.