Stabiilien mallien toteutustekniikkaa

- Tavoiteena laskea mahdollisimman tehokkaasti logiikkaohjelman P annetut kriteerit täyttävä(t) stabiili(t) malli(t).
- Täysiin joukkoihin perustuvan stabiilien mallien karakterisoinnin perusteella hakuavaruus muodostuu olennaisesti joukon $\mathrm{NA}(P)$ (ohjelmassa P esiintyvät not-literaalit) osajoukoista.
- Rakennetaan asteittain stabiileja malleja rajaavaa osittaismallia (literaalijoukko FS) ja pyritään karsimaan hakuavaruutta.
  (i) Malleja rajaavat oletukset tehdään yksi kerrallaan.
  (ii) Jokaisessa hakupisteessä lasketaan approksimaatio kaikille stabiileille malleille, jotka toteuttavat senhetkiset oletukset ja tutkitaan, onko mahdollisesti saavutettu konflikti.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

☞ Saadaan Davis-Putnam -menetelmää muistututtava algoritmi stabiilien mallien laskemista varten.

- Toteutustekniikkaa analysoidaan mm. Simonsin väitöskirjassa:
  P. Simons [2000]: *Extending and Implementing the Stable Model Semantics*
- Lisäksi on toteutettu muuttujia sisältävien ohjelmien instantiointi:
  T. Syrjänen [1999]: *Implementation of Local Grounding for Logic Programs with Stable Model Semantics*
- Ohjelmat `smodels` ja `lparse` ovat saatavilla verkossa osoitteella

  `http://www.tcs.hut.fi/pub/smodels/`

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Approksimointiperiaatteet

- Laskettavat stabiilit mallit rajataan literaalijoukolla FS:
  - (i) Jos $A \in FS$, niin $A \in M$ laskettaville stabiileille malleille M.
  - (ii) Jos $\mathrm{not}\ A \in FS$, niin $A \notin M$ laskettaville stabiileille malleille M (näitä vastaaville täysille joukoille F pätee $\mathrm{not}\ A \in F$).
- Tämä ehdot voidaan ymmärtää ohjelman P stabiiliin malliin M ja literaalijoukon FS välisenä *yhteensopivuutena*.

Esimerkki. Tarkastellaan normaalia logiikkaohjelmaa

$$P = \{\ A \leftarrow \mathrm{not}\ B,\ B \leftarrow \mathrm{not}\ A,\ C \leftarrow \mathrm{not}\ D,$$
$$D \leftarrow \mathrm{not}\ C,\ E \leftarrow \mathrm{not}\ F,\ F \leftarrow \mathrm{not}\ E\ \}.$$

Nyt esim. literaalijoukko $FS = \{A, \mathrm{not}\ C\}$ on yhteensopiva ohjelman P stabiilien mallien $M_1 = \{A, D, E\}$ ja $M_2 = \{A, D, F\}$ kanssa.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

- Approksimointi perustuu stabiilien mallien ala- ja ylärajaan:
  1. *Alaraja* $\mathrm{LB}(P, FS) \supseteq FS$ on literaalijoukko, joka on yhteensopiva jokaisen literaalijoukon FS kanssa yhteensopivan ohjelman P stabiilin mallin $M \subseteq \mathrm{HB}(P)$ kanssa.
  2. *Yläraja* $\mathrm{UB}(P, FS) \subseteq \mathrm{HB}(P)$ on atomilauseiden joukko, joka sisältää jokaisen literaalijoukon FS kanssa yhteensopivan ohjelman P stabiilin mallin $M \subseteq \mathrm{HB}(P)$.
- Approksimaatio **expand**(P, FS) on pienin joukko FS', joka sisältää FS:n ja on suljettu seuraavien sääntöjen suhteen:
  - (i) Jos literaali $L \in \mathrm{LB}(P, FS')$, niin $L \in FS'$.
  - (ii) Jos atomi $A \notin \mathrm{UB}(P, FS')$, niin $\mathrm{not}\ A \in FS'$.
- Approksimaatio **expand**(P, FS) saadaan laskemalla ala- ja ylärajoja toistuvasti ja täydentämällä literaalijoukkoa FS (kunnes saavutetaan kiintopiste periaatteiden (i) ja (ii) suhteen).

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Approksimaatioiden toteutus

- Tavoitteena on, että alaraja $\mathrm{LB}(P, FS)$ ja yläraja $\mathrm{UB}(P, FS)$ voidaan laskea lineaarisessa ajassa.
- Vähintään yhtä tarkat rajat kuin WF-mallilla.
- Hakutilanne (osittaismalli FS) hyödynnettävä tehokkaasti.
- Oletuksilla FS vielä (mahdollisesti) käytettävissä olevat säännöt:

$$P_{FS} = \{\ H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \mathrm{not}\ B_{m+1} \wedge \cdots \wedge \mathrm{not}\ B_n \in P \mid FS \cap \{\mathrm{not}\ B_1, \ldots, \mathrm{not}\ B_m, B_{m+1}, \ldots, B_n\} = \emptyset\ \}.$$

- Määritellään literaalin L komplementti $\overline{L}$ normaaliin tapaan:

$$\overline{A} = \mathrm{not}\ A \text{ ja } \overline{\mathrm{not}\ A} = A.$$

- Nyt $P_{FS} = \{H \leftarrow L_1 \wedge \cdots \wedge L_n \in P \mid FS \cap \{\overline{L_1}, \ldots, \overline{L_n}\} = \emptyset\}$.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Alaraja

Määritelmä. Alaraja $\mathrm{LB}(P, FS)$ on pienin joukko FS', joka sisältää joukon FS ja on suljettu seuraavien periaatteiden suhteen:

S1: Jos $H \leftarrow L_1 \wedge \cdots \wedge L_n \in P_{FS'}$ ja $\{L_1, \ldots, L_n\} \subseteq FS'$, $H \in FS'$.

S2: Jos atomi H ei esiinny minkään ohjelman $P_{FS'}$ säännön seurauksena, niin $\mathrm{not}\ H \in FS'$.

S3: Jos $H \in FS'$ on täsmälleen yhden säännön $H \leftarrow L_1 \wedge \cdots \wedge L_n \in P_{FS'}$ seurauksena, $\{L_1, \ldots, L_n\} \subseteq FS'$.

S4: Jos $\mathrm{not}\ H \in FS'$, sääntö $H \leftarrow L_1 \wedge \cdots \wedge L_n \in P_{FS'}$ ja $\{L_1, \ldots, L_{i-i}, L_{i+1}, \ldots, L_n\} \subseteq FS'$, niin $\overline{L_i} \in FS'$.

S5: Jos jollekin atomille H sekä $H \in FS'$ että $\mathrm{not}\ H \in FS'$, niin kaikki literaalit kuuluvat joukkoon FS'.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Tarkastellaan ohjelmaa

$$P_1 = \{\ A \leftarrow \text{not } B \text{ (R1)}, C \leftarrow A \text{ (R2)},$$
$$B \leftarrow A \wedge \text{not } C \wedge \text{not } D \text{ (R3)}, D \leftarrow C \wedge \text{not } E \text{ (R4)}\ \}.$$

➤ Alarajan $\mathrm{LB}(P_1, \{\text{not } B\})$ laskenta:

Literaalit	Periaate	Käytettävissä olevat säännöt
not B	oletus	R1, R2, R3, R4
A	S1	R1, R2, R3, R4
C	S1	R1, R2, R4
not E	S2	R1, R2, R4
D	S1	R1, R2, R4

➤ Approksimaatio **expand**$(P_1, \{\text{not } B\}) = \{\text{not } B, A, C, \text{not } E, D\}$ määrää jo stabiilin mallin $\{A, C, D\}$ (johon atomi B ei kuulu).

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Tarkastellaan edelleen ohjelmaa

$$P_1 = \{\ A \leftarrow \text{not } B \text{ (R1)}, C \leftarrow A \text{ (R2)},$$
$$B \leftarrow A \wedge \text{not } C \wedge \text{not } D \text{ (R3)}, D \leftarrow C \wedge \text{not } E \text{ (R4)}\ \}.$$

➤ Alarajan $\mathrm{LB}(P_1, \{B\})$ laskenta:

Literaalit	Periaate	Käytettävissä olevat säännöt
B	oletus	R2, R3, R4
A, not C, not D	S3	R2, R3
C	S1	R2
kaikki literaalit	S5	

➤ Täten myös **expand**$(P_1, \{B\})$ antaa kaikki literaalit

☞ ohjelmalla P_1 ei ole stabiilia mallia M siten, että $B \in M$.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Yläraja

Määritelmä. Yläraja $\mathrm{UB}(P, FS)$ on pienin mallin ohjelmalle P', joka saadaan poistamalla not-literaalit ohjelmasta P_{FS}.

Esimerkki. Tarkastellaan ohjelmaa

$$P_2 = \{\ A \leftarrow \mathrm{not}\ B,\ B \leftarrow \mathrm{not}\ A,\ C \leftarrow \mathrm{not}\ A,$$
$$D \leftarrow \mathrm{not}\ C,\ E \leftarrow \mathrm{not}\ D\ \}$$

Lasketaan ohjelmalle P_2 esim. seuraavat ylärajat:

1. $\mathrm{UB}(P_2, \emptyset) = \{A, B, C, D, E\}$
   (koska lisäksi myös $\mathrm{LB}(P_2, \emptyset) = \emptyset$, saadaan **expand**$(P_2, \emptyset) = \emptyset$).
2. $\mathrm{UB}(P_2, \{A\}) = \{A, D, E\}$.
3. $\mathrm{UB}(P_2, \{\mathrm{not}\ A\}) = \{A, B, C, D, E\}$.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Lasketaan samaiselle ohjelmalle

$$P_2 = \{\ A \leftarrow \mathrm{not}\ B,\ B \leftarrow \mathrm{not}\ A,\ C \leftarrow \mathrm{not}\ A,$$
$$D \leftarrow \mathrm{not}\ C,\ E \leftarrow \mathrm{not}\ D\ \}$$

approksimaatiot **expand**$(P_2, \{A\})$ ja **expand**$(P_2, \{\mathrm{not}\ A)$.

1. $\mathrm{LB}(P_2, \{A\}) = \{A, \mathrm{not}\ B, \mathrm{not}\ C, D, \mathrm{not}\ E\} =$ **expand**$(P_2, \{A\})$.
2. $\mathrm{LB}(P_2, \{\mathrm{not}\ A\}) = \{\mathrm{not}\ A, B, C, \mathrm{not}\ D, E\} =$
   expand$(P_2, \{\mathrm{not}\ A\})$.

Esimerkki. Ohjelmalle $P_3 = \{A \leftarrow \mathrm{not}\ B, B \leftarrow B\}$ alaraja $\mathrm{LB}(P_3, \emptyset) = \emptyset$, mutta yläraja $\mathrm{UB}(P_3, \emptyset) = \{A\}$.

Näin voidaan päätellä $\mathrm{not}\ B$ (kts. **expand**). Tästä seuraa, että **expand**$(P_3, \emptyset) = \{A, \mathrm{not}\ B\}$, koska $\mathrm{LB}(P_3, \{\mathrm{not}\ B\}) = \{A, \mathrm{not}\ B\}$.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Ala- ja ylärajojen toteuttaminen

- Ala- ja ylärajoille voidaan antaa lineaarisen ajan toteutukset (käyttämällä Dowling-Gallier -tyyppisiä tietorakenteet).
- Tarkkuus on vähintään sama kuin WF-mallilla: **expand**$(P, \emptyset)$ antaa ohjelman P WF-mallin.
- Hakutilanne (osittaismalli FS) hyödynnetään dynaamisesti.

Esimerkki. Ohjelman $P = \{A \leftarrow \text{not } B,\ B \leftarrow \text{not } A,\ A \leftarrow B\}$ tapauksessa approksimaatioksi **expand**$(P, \{B\})$ saadaan $\{A, B, \text{not } A, \text{not } B\}$.

Ohjelman $P \cup \{B\}$ WF-malli on $\{A, B\}$ (konflikti ei tule esille)!

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Stabiilien mallien laskenta

function smodels(P, FS, φ): boolean;
FS' := **expand**(P, FS);
if conflict(P, FS') returns true **then** return false
else if $\text{NA}(P)$ is covered by FS' **then** return **test**(P, FS', φ)
else
  χ := **choose**(P, FS');
  if smodels$(P, FS' \cup \{\chi\}, \varphi)$ returns true **then**
    return true
  else return **smodels**$(P, FS' \cup \{\overline{\chi}\}, \varphi)$
  end if
end if

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Funktioiden tehtävät:

- Funktiota **test** käytetään löytyneiden mallien valintaan (parametri φ antaa valintakriteerit).
- Funktio **conflict**(P, FS) tarkistaa, sisältääkö saatu approksimaatio FS konfliktin ($\{A, \text{not } A\} \subseteq FS$ jollekin $A \in \text{HB}(P)$).
- Funktio **choose** toteuttaa hakuheuristiikan (paluuarvona literaali).
- Funktio **smodels** suorittaa *fokusoitua mallinetsintää*:

  smodels(P, FS, φ) palauttaa arvon true
  $\iff$ $\exists$ P:n stabiili malli $M = \text{Dcl}(P, \{\text{not } A \mid \text{not } A \in FS'\})$,
  joka on yhteensopiva FS:n kanssa ($FS \subseteq FS'$) ja
  jolle **test**(P, FS', φ) palauttaa arvon true.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Tutkitaan, kuuluuko atomi E ohjelman P kaikkiin stabiileihin malleihin etsimällä vastamalli, johon E ei kuulu.

Aloitetaan osittaismallista $FS = \{\text{not } E\}$
(valintafunktio **test** palauttaa aina arvon true).

$$
\begin{aligned}
P = \{\ & A \leftarrow \text{not } B, \\
& B \leftarrow \text{not } A, \\
& C \leftarrow A \wedge \text{not } B, \\
& C \leftarrow B \wedge \text{not } A, \\
& D \leftarrow \text{not } C, \\
& E \leftarrow \text{not } D\ \}
\end{aligned}
\qquad
\begin{array}{c}
\text{not } E \\
D \\
\text{not } C \\
\begin{array}{c|c} A & \text{not } A \\ \text{not } B & B \\ C & C \\ \times & \times \end{array}
\end{array}
$$

☞ Kyllä kuuluu.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Esimerkki. Etsitään ohjelman

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

stabiilit mallit (alussa $FS = \emptyset$ ja **test** palauttaa aina true):

A	not A
not D	B
C	D
not B	not C

☞ Stabiilit mallit $M_1 = \text{Dcl}(P, \{\text{not } B, \text{not } D\}) = \{A, C\}$ ja $M_2 = \text{Dcl}(P, \{\text{not } A, \text{not } C\}) = \{B, D\}$.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Heuristiikka

- Lookahead-tekniikka: jokaiselle atomille $A \in \text{NA}(P)$, johon osittaismalli FS ei ota kantaa ($A \notin FS$ ja not $A \notin FS$), lasketaan
  $FS_1 =$ **expand**$(P, FS \cup \{A\})$ ja
  $FS_2 =$ **expand**$(P, FS \cup \{\text{not } A\})$
- Saadaan tarkempi alaraja:
  Jos FS_1 sisältää konfliktin, lisätään not A joukkoon FS ja jos FS_2 sisältää konfliktin, lisätään A joukkoon FS.
- FS_1 and FS_2 antavat arvion jäljellä olevasta hakuavaruudesta.
- Heuristiikka: valitaan atomi, jolla on pienin arvioitu jäljellä oleva hakuavaruus.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Arviointia

- Logiikkaohjelmat ja stabiilit mallit ilmaisuvoimainen rajoiteohjelmointiparadigma.
  (Esim. Boolen rajoitteet/lauselogiikka erikoistapaus).
- Hakuavaruuden rakenne poikkeaa lauselogiikan toteutuvuusongelmasta:
  (i) Vain not-literaalit lisäävät hakuavaruutta.
  (ii) Säännöt suunnattuja: Kerrostuneisuus rajaa hakuavaruutta.
  ☞ Enemmän vapautta tietämyksen esittämiseen
- Toteutustekniikat parantuneet ratkaisevasti: mm. edellä kuvatun algoritmin tilavaativuus on lineaarinen ohjelman kokoon nähden.

© 2001 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio