Toteutustekniikkaa

[Niemelä, Simons, 1999] `smodels` ja [Syrjänen, 1999] `lparse`

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

- Käytetään kompaktia karakterisointia (täydet joukot)
  ☞ hakuavaruus: not-literaalit
- Bottom-up haku & hakuavaruuden karsinta:
  (i) Oletukset tutkittavasta mallista tehdään yksi kerrallaan.
  (ii) Ennen seuraavan oletuksen teko lasketaan approksimaatio kaikille stabiileille malleille, jotka toteuttavat annetut oletukset ja tutkitaan ollaanko jo konfliktissa.

☞ Davis-Putnam -menetelmän tyylinen tekniikka

- Algoritmi säilyttää approksimaatiota stabiileista malleista literaalijoukkona (osittaismallina) FS:
  (i) Jos $A \in FS$, A kuuluu tutkittaviin stabiileihin malleihin.
  (ii) Jos $\mathrm{not}\ A \in FS$, A ei kuulu tutkittaviin stabiileihin malleihin ($\mathrm{not}\ A$ kuuluu täyteen joukkoon).
- Approksimointi perustuu stabiilien mallien ala- ja ylärajaan:
  $\mathrm{LB}(P, FS)$ on literaalijoukko, jonka jokainen FS:n toteuttava ohjelman P stabiili malli toteuttaa.
  $\mathrm{UB}(P, FS)$ on atomilauseiden joukko, joka sisältää jokaisen ohjelman P stabiilin mallin, joka toteuttaa FS.
- Approksimaatio **expand**(P, FS) on pienin joukko FS', joka sisältää FS ja on suljettu seuraavien sääntöjen suhteen:
  (i) jos literaali $L \in \mathrm{LB}(P, FS')$, $L \in FS'$ ja
  (ii) jos atomi $A \notin \mathrm{UB}(P, FS')$, $\mathrm{not}\ A \in FS'$.

Approksimaatioiden toteutus

- Tavoite: ala- ja yläraja laskettavissa lineaarisessa ajassa.
- Vähintään yhtä tarkat rajat kuin WF-mallilla.
- Hakutilanne (osittaismalli) hyödynnettävä tehokkaasti.
- Oletuksilla FS vielä käytettävissä olevat säännöt:
  $P_{FS} = \{H \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \mathrm{not}\ B_{m+1} \wedge \cdots \wedge \mathrm{not}\ B_n \in P \mid \{\mathrm{not}\ B_1, \ldots, \mathrm{not}\ B_m, B_{m+1}, \ldots, B_n\} \cap FS = \emptyset\}$
- $\overline{L}$ on literaalin L:n komplementti:
  $\overline{\mathrm{not}\ A} = A$ ja $\overline{A} = \mathrm{not}\ A$.

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

1. Jos $H \leftarrow L_1 \wedge \cdots \wedge L_n \in P_{FS'}$ ja $\{L_1, \ldots, L_n\} \subseteq FS'$, $H \in FS'$.
2. Jos atomi H ei ole päänä ohjelmassa $P_{FS'}$, $\mathrm{not}\ H \in FS'$.
3. Jos $H \in FS'$ on päänä vain yhdessä säännössä $H \leftarrow L_1 \wedge \cdots \wedge L_n \in P_{FS'}$, $\{L_1, \ldots, L_n\} \subseteq FS'$.
4. Jos $\mathrm{not}\ H \in FS'$, $H \leftarrow L_1 \wedge \cdots \wedge L_n \in P_{FS'}$ ja $\{L_1, \ldots, L_{i-i}, L_{i+1}, \ldots, L_n\} \subseteq FS'$, $\overline{L_i} \in FS'$.

$\mathrm{UB}(P, FS)$: pienin malli ohjelmaklausuulijoukolle, joka saadaan poistamalla not-literaalit ohjelmasta P_{FS}.

Esimerkki. a) P :

$p \leftarrow \text{not } q$

$r \leftarrow \text{not } p$

$q \leftarrow p \land \text{not } r \land \text{not } s$

$s \leftarrow \text{not } t \land r$

$\text{LB}(P, \{\text{not } q\}) =$
$\{\text{not } q, p, \text{not } r, s, \text{not } t, r\}$
(konflikti r:n suhteen)

b) P' :

$p \leftarrow q$

$q \leftarrow p$

$p \leftarrow \text{not } r$

$r \leftarrow \text{not } s$

$s \leftarrow \text{not } p \land \text{not } q$

$\text{LB}(P', \{r\}) = \{r, \text{not } s\}$

$\text{UB}(P', \{r, \text{not } s\}) = \{r, s\}$

$\text{LB}(P', \{r, \text{not } s, \text{not } p, \text{not } q\}) =$
$\{r, \text{not } s, \text{not } p, \text{not } q, s, \text{not } r\}$

$\text{UB}(P', \{r, \text{not } s, \text{not } p, \text{not } q, s, \text{not } r\})$
$= \{s\}$, joten **expand**$(P, \{r\}) =$
$\{r, \text{not } s, \text{not } p, \text{not } q, s, \text{not } r\}$

- Ala- ja ylärajoille voidaan antaa lineaarisen ajan toteutukset (Dowling-Gallier -tyyppiset tietorakenteet).
- Tarkkuus vähintään sama kuin WF-mallilla:
  Jos oletuksia ei ole, **expand**$(P, \{\})$ antaa ohjelman P WF-mallin.
- Hakutilanne (oletukset FS) hyödynnetään dynaamisesti.
  Esim. ohjelmalle P
  $p \leftarrow \text{not } q$
  $q \leftarrow \text{not } p$
  $p \leftarrow q$
  expand$(P, \{q\})$ palauttaa $\{q, p, \text{not } q, \text{not } p\}$.
  Huom! Ohjelman $P \cup \{q\}$ WF-malli on $\{q, p\}$
  (konflikti ei tule esille).

Algoritmi

function smodels(P, FS, φ)
$FS' :=$ **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
    $\chi :=$ **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

- Funktiota **test** käytetään löytyneen mallin testaukseen (φ on testiparametri).
- Funktio **conflict** tarkistaa, sisältääkö saatu approksimaatio konfliktin ($\{A, \text{not } A\} \subseteq FS$).
- Funktio **choose** toteuttaa hakuheuristiikan.
- **smodels** suorittaa *fokusoitua mallinetsintää*:
  smodels(P, FS, φ) palauttaa true joss on olemassa P:n stabiili malli M, joka toteuttaa FS ja **test**(P, FS', φ) palauttaa true missä $M = \text{Dcl}(P, FS')$.

Esimerkki.

a) Tutkitaan, kuuluuko t P:n kaikkiin stabiileihin malleihin etsimällä vastamalli, jossa t ei ole. Aloitetaan $FS = \{\mathrm{not}\ t\}$ ja **test** palauttaa aina true.

$p \leftarrow \mathrm{not}\ q$
$q \leftarrow \mathrm{not}\ p$
$r \leftarrow p$
$r \leftarrow q$
$s \leftarrow \mathrm{not}\ r$
$t \leftarrow \mathrm{not}\ s$

$\mathrm{not}\ t$	
s	
$\mathrm{not}\ r$	
p	$\mathrm{not}\ p$
r	q
$\times$	r
	$\times$

Vastaus: kyllä

b) Etsitään ohjelman P

$p \leftarrow \mathrm{not}\ q \wedge r$
$q \leftarrow \mathrm{not}\ p$
$r \leftarrow \mathrm{not}\ s$
$s \leftarrow \mathrm{not}\ p$

stabiilit mallit: (aluksi $FS = \{\}$)

p	$\mathrm{not}\ p$
$\mathrm{not}\ s$	q
r	s
$\mathrm{not}\ q$	$\mathrm{not}\ r$

Löytyi $\mathrm{Dcl}(P, \{p, \mathrm{not}\ s, r, \mathrm{not}\ q\})$ ja $\mathrm{Dcl}(P, \{\mathrm{not}\ p, q, s, \mathrm{not}\ r\})$

Heuristiikka

- lookahead-tekniikka:
  jokaiselle $A \in \mathrm{NA}(P)$, jota FS ei peitä, lasketaan
  $FS_1 = \textbf{expand}(P, FS \cup \{A\})$ ja
  $FS_2 = \textbf{expand}(P, FS \cup \{\mathrm{not}\ A\})$
- Saadaan tarkempi alaraja:
  Jos FS_1 sisältää konfliktin, lisätään $\mathrm{not}\ A$ 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.

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.
  Esimerkiksi lineaarinen tilavaatiivuus.