Príklady na stabilné modely

Množinu všetkých stabilných modelov programu P označme $\mathrm{SM}(P)$.

Úloha 1. Nájdite normálny logický program, pre ktorý platí:

- $\mathrm{SM}(P) = \emptyset$
- $\mathrm{SM}(P) = \{\emptyset\}$

Úloha 2. Napíšte NLP P, kde $\mathrm{SM}(P) = \{\{padla(i)\} \mid 1 \leq i \leq 6\}$ bude zodpovedať možným hodom normálnej hracej kocky.

Gelfond-Lifschitzovu transformáciu programu P podľa I budeme značiť P^I.

Úloha 3. Nech P je ľubovoľný normálny logický program a I ľubovoľná herbrandovská interpretácia. Dokážte, že P^I má najmenší model.

Úloha 4. Majme normálny logický program P a ľubovoľné stabilné modely $A, B \in \mathrm{SM}(P)$. Dokážte, že ak $A \subseteq B$, potom $A = B$.

Úloha 5. Majme normálny logický program P. Ak je P stratifikovaný, potom má prave jeden stabilný model ($|\mathrm{SM}(P)| = 1$).
Návod: Predpokladajte, že $P_1, \ldots, P_n$, kde $n \in \mathbb{N}$ je nejaké prirodzené číslo, je stratifikácia programu P. Označme model (získaný iteratívnou konštrukciou podľa stratifikácie) programu P_i ako S_{P_i}. Štandardný model programu P je teda S_{P_n} a označme ho len ako S. $S \in \mathrm{SM}(P)$ dokážeme tak, že ukážeme postupne $T_{PS} \uparrow i \subseteq S$ pre všetky $i \in \mathbb{N}$. Potom ukážeme, že $lfp(T_{PS}) \supseteq S_{P_i}$ pre $1 \leq i \leq n$. Nakoniec jedinečnosť ($|\mathrm{SM}(P)| = 1$) dokážeme z predošlého a pomocou úlohy 4.

Predchádzajúci výsledok znamená, že stabilný model je zovšeobecnením štandardného modelu.

Úloha 6. Čo uvedený program vypočíta v predikáte $result(X)$ v prípade, že $less(X, Y)$ je

- lineárne usporiadanie (každá dvojica prvkov je porovnateľná)?
- čiastočné usporiadanie (nejaká dvojica prvkov môže byť neporovnateľná)?

$$\begin{aligned} not\_result(X) &\leftarrow object(X), object(Y), less(Y, X). \\ result(X) &\leftarrow object(X), \sim not\_result(X). \end{aligned}$$

Úloha 7. Navrhnite takú transformáciu rozšíreného logického programu P do defaultovej teórie (E, D), aby platilo $\mathrm{EX}(E, D) = \{\mathrm{Cn}_{FOL}(S) \mid S \in \mathrm{SM}(P)\}$, kde $\mathrm{EX}(E, D)$ je množina všetkých defaultových extenzii defaultovej teórie (E, D).

Príklad 1. Majme nasledovný scenár.

1. Európania sú zvyčajne civilizovaní.

2. Športoví chuligáni väčšinou nie sú civilizovaní, pokiaľ nie sú vzdelaní.

3. Ani vzdelaní športoví chuligáni nie sú civilizovaní, ak sú opití.

Jedna z možných formalizácii v jazyku rozšíreného logického programu je nasledujúci program P:

1. $civilized(X) \leftarrow european(X), \sim\neg civilized(X)$

2. $\neg civilized(X) \leftarrow hooligan(X), \sim educated(X)$

3. $\neg civilized(X) \leftarrow hooligan(X), educated(X), drunk(X)$

Je táto formalizácia dobrá? To závisí od toho, čo sme skutočne chceli povedať.

Uvažujme nasledovné fakty

$$F = \{\neg european(peter).\ civilized(peter).\ \neg educated(peter).\ hooligan(peter).\}$$

„Problém"je, že program $P \cup F$ nemá (konzistentný) stabilný model (odvodíme $civilized(peter)$ aj $\neg civilized(peter)$). V prípade, že program P formalizuje *presne* to, čo sme chceli, neexistenciu stabilných modelov programu $P \cup F$ nepovažujeme za problém a znamená to len, že fakty F nemajú vzhľadom na náš model sveta (program P) zmysel a preto ich netreba uvažovať. V prípade, že fakty F majú v našom scenári zmysel, formalizácia P nie je adekvátna a treba ju zmeniť, aby bola odolná voči takýmto výnimkám (v literatúre sa tomu hovorí *elaboration tolerance*). V tomto prípade by mohla byť zmena jednoduchá: stačí pravidlo 2 nahradiť pravidlom

$$\neg civilized(X) \leftarrow hooligan(X), \sim educated(X), \sim\neg civilized(X)$$

Uvažujme teraz tieto fakty

$$G = \{european(peter).\ educated(peter).\ hooligan(peter).\}$$

Program $P \cup G$ má jediný stabilný model $G \cup \{civilized(peter)\}$. Niekedy môžeme preferovať opatrnejšie odvodenie, prípadne môžeme mať preferencie. Preferencie nemusia byť definované len explicitne medzi pravidlami, ale aj umožnením odvodzovania na základe defaultovej znalosti. Ak by sme pravidlo 3 nahradili pravidlom

$$\neg civilized(X) \leftarrow hooligan(X), educated(X), \sim\neg drunk(X)$$

jediným stabilným modelom by už bol $G \cup \{\neg civilized(peter)\}$.

Príklad 2. Uvažujme variáciu programu o chuligánoch. Majme program P

1. $civilized(X) \leftarrow european(X), \sim ab(X)$

2. $ab(X) \leftarrow hooligan(X), \sim educated(X)$

3. $ab(X) \leftarrow hooligan(X), drunk(X)$

a fakty $\{drunk(peter)\}$. P je normálny logický program. Sémantika NLP je dvojhodnotová, teda o každom atóme vieme povedať či je alebo nie je pravdivý (robí CWA). V NLP vyplývanie vzhľadom na daný stabilný model S je zadefinované ako $S \models_{\text{NLP}} \neg A$ ak $A \notin S$, kde A je atóm. Teda platí $P \models_{NLP} \neg civilized(peter)$.

Avšak NLP sú špeciálnym prípadom rozšírených logických programov (ELP), ktoré CWA nepredpokladajú. V ELP sú stabilné modely množiny literálov, takže pracujeme s tromi hodnotami (*true, false, unknown*) a vyplývanie vzhľadom na daný stabilný model S je zadefinované ako $S \models_{\text{ELP}} L$ ak $L \in S$, kde L je klasický literál. Teda $P \not\models_{\text{ELP}} \neg civilized(peter)$, aj $P \not\models_{\text{ELP}} civilized(peter)$. Teda $civilized(peter)$ má hodnotu unknown.

Úloha 8. Doplňte program P z príkladu 2 o také pravidlá, aby sa vyplývanie $\models_{\text{ELP}}$ zhodovalo s vyplývaním $\models_{\text{NLP}}$. Teda, pravidlami explicitne vyjadrite CWA v ELP.