FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Knowledge-Based Systems Group

Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-13), Bellevue, Washington, USA, July 14-18, 2013

Liberal Safety for Answer Set Programs with External Sources

Thomas Eiter Michael Fink Thomas Krennwallner Christoph Redl
{eiter,fink,tkren,redl}@tuwien.ac.at

1. Motivation

- HEX-programs extend ASP by external sources
- Rule bodies may contain external atoms of the form

$$\&p[q_1, \ldots, q_k](t_1, \ldots, t_l),$$

  p . . . external predicate name
  q_i . . . predicate names or constants: $\tau(\&p, i) \in \{\mathbf{pred}, \mathbf{const}\}$
  t_j . . . terms
  Semantics:
  $1 + k + l$-ary Boolean oracle function $f_{\&p}$:
  $\&p[q_1, \ldots, q_k](t_1, \ldots, t_l)$ is true under assignment $\mathbf{A}$
  iff $f_{\&p}(\mathbf{A}, q_1, \ldots, q_k, t_1, \ldots, t_l) = 1$.
- Traditional safety not sufficient due to value invention
- Current notion of strong safety is unnecessarily restrictive

Example

$$\Pi = \begin{Bmatrix} r_1: t(a). & r_3: s(Y) \leftarrow t(X), \&cat[X, a](Y). \\ r_2: dom(aa). & r_4: t(X) \leftarrow s(X), dom(X). \end{Bmatrix}$$

Goal:

- New more liberal safety criteria
- Still guarantee finite groundability
- Future extensibility

Main result:

- Flexible framework which subsumes other notions and allows for combination of syntactic and semantic safety criteria

2. Grounding Operator

We use the following canonical grounding operator:

$$G_\Pi(\Pi') = \bigcup_{r \in \Pi}\{r_\theta \mid \mathbf{A} \subseteq \mathcal{A}(\Pi'), \mathbf{A} \not\models \bot, \mathbf{A} \models B^+(r_\theta)\},$$

where $\mathcal{A}(\Pi') = \{\mathbf{T}a, \mathbf{F}a \mid a \in A(\Pi')\} \setminus \{\mathbf{F}a \mid a \leftarrow . \in \Pi\}$
and r_θ is the instance of r under variable substitution $\theta: \mathcal{V} \to \mathcal{C}$.

For liberally domain-expansion safe programs, the fixpoint of this operator is reached after finitely many steps and preserves all answer sets.

Example

Program Π:

$$r_1: s(a). \quad r_2: dom(ax). \quad r_3: dom(axx).$$
$$r_4: s(Y) \leftarrow s(X), \&cat[X, x](Y), dom(Y).$$

Least fixpoint of G_Π:

$$r'_1: s(a). \quad r'_2: dom(ax). \quad r'_3: dom(axx).$$
$$r'_4: s(ax) \leftarrow s(a), \&cat[a, x](ax), dom(ax).$$
$$r'_5: s(axx) \leftarrow s(ax), \&cat[ax, x](axx), dom(axx).$$

3. Two Key Concepts

- A term is bounded if $G_\Pi(\Pi')$ contains only finitely many substitutions for this term
- An attribute (=argument position of a predicate or external predicate) is (liberally) de-safe if $G_\Pi(\Pi')$ contains only finitely many values at this attribute position

A program is de-safe iff all its attributes are de-safe

4. How to Check Safety

1. Start with empty set of bounded terms B_0 and de-safe attributes S_0
2. For all $n \geq 0$ until B_n and S_n did not change
   a. Declare additional terms as bounded $\Rightarrow B_{n+1}$
      (assuming that B_n are bounded and S_n are de-safe)
   b. Identify additional de-safe attributes $\Rightarrow S_{n+1}$
      (assuming that B_{n+1} are bounded and S_n are de-safe)

Identification of bounded terms in 2a by term bounding functions (TBFs)
Concrete safety criteria can be plugged in by specific TBF $b(\Pi, r, S, B)$
$\Rightarrow$ TBFs are easily exchangeable but must fulfill certain preconditions

5. Example: Syntactic Term Bounding Function

$t \in b_{syn}(\Pi, r, S, B)$ iff

(i) t is a constant in r; or
(ii) there is an ordinary atom $q(s_1, \ldots, s_{ar(q)}) \in B^+(r)$ s.t. $t = s_j$, for some $1 \leq j \leq ar(q)$ and $q \upharpoonright j \in S$; or
(iii) for some external atom $\&g[\vec{X}](\vec{Y}) \in B^+(r)$, we have that $t = Y_i$ for some $Y_i \in \vec{Y}$, and for each $X_i \in \vec{X}$,

$$\begin{cases} X_i \in B, & \text{if } \tau(\&g, i) = \mathbf{const}, \\ X_i \upharpoonright 1, \ldots, X_i \upharpoonright ar(X_i) \in S, & \text{if } \tau(\&g, i) = \mathbf{pred}. \end{cases}$$

6. Key Propositions

Proposition 1:
If $b_i(\Pi, r, S, B)$, $1 \leq i \leq \ell$, are TBFs, then $\bigcup_{1 \leq i \leq \ell} b_i(\Pi, r, S, B)$ is a TBF.
$\Rightarrow$ Easy combination of multiple TBFs

Proposition 2:
If Π is a de-safe program, then $G^\infty_\Pi(\emptyset)$ is finite.

Proposition 3:
A de-safe program Π is finitely restrictable and $G^\infty_\Pi(\emptyset) \equiv^{pos} \Pi$.
$\Rightarrow$ Operator G is a witness for finite groundability

Note: The results hold for any TBF!

7. Advantages

- **Strictly more liberal** than strongly safe [Eiter et al., 2006], VI- [Calimeri et al., 2007] and ω-restricted [Syrjänen, 2001] programs.
- **Modularity** of the approach
- **Extensible** due to easy combination of multiple TBFs

8. References

- Calimeri, F., Cozza, S., and Ianni, G. (2007).
  External Sources of Knowledge and Value Invention in Logic Programming
  Ann. Math. Artif. Intell. 50(3–4):333–361.
- Eiter, T., Ianni, G., Schindlauer, R., and Tompits, H. (2006).
  Effective Integration of Declarative Rules with External Evaluations for Semantic-Web Reasoning
  ESWC'06 volume 4011, pages 273–287.
- Syrjänen, T. (2001).
  Omega-restricted logic programs, LPNMR'01, volume 2173 of LNCS, pages 267–279.
- Zantema, H. (1994).
  Termination of term rewriting: Interpretation and type elimination, J. Symb. Comp., 17(1):23–50.

FWF
Der Wissenschaftsfonds.