Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14), Quèbec City, Canada, July 27-31, 2014

Exploiting Support Sets for Answer Set Programs with External Evaluations

Thomas Eiter   Michael Fink   Christoph Redl   Daria Stepanova

{eiter,fink,redl,dasha}@kr.tuwien.ac.at

1. Motivation

- **HEX-programs** extend ASP by external sources
- DL-program $\Pi = \langle \mathcal{O}, P \rangle$, ontology+rules (special case of HEX-programs)

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ Driver \sqsubseteq \neg Customer & (4)\ worksIn(d1, r3) \\ (2)\ \exists.worksIn \sqsubseteq Driver & (5)\ worksIn(d1, r4) \\ (3)\ ECarDriver \sqsubseteq Driver & (6)\ ECarDriver(d1) \end{array} \right\}$$

$$P = \left\{ \begin{array}{l} (6)\ isIn(c1, r2);\ \ (7)\ isIn(d1, r2);\ \ (8)\ needsTo(c1, r3)\ \ (9)\ goTo(d1, r4); \\ (10)\ cust(X) \leftarrow isIn(X, Y), not\ \mathbf{DL}[worksIn \uplus goTo; \neg Customer](\mathbf{X}); \\ (11)\ driver(X) \leftarrow not\ customer(X), isIn(X, Y); \\ (12)\ drives(X, Y) \leftarrow driver(X), customer(Y), isIn(X, Z), \\ \qquad\qquad isIn(Y, Z), not\ omit(X, Y); \\ (13)\ omit(X, Y) \leftarrow needsTo(Y, Z), not\ \mathbf{DL}[; worksIn](X, Z), \\ \qquad\qquad drives(X, Y), \mathbf{DL}[Driver \uplus driver; ECarDriver](X) \end{array} \right\}$$

- **Evaluation of HEX-programs**: **multiple calls to external sources** are expensive!

- **Aim of this work:** avoid multiple calls
- **Contributions:**
  - (Non-)ground support sets as optimization means
  - Application examples:
    - DL-programs ($DL\text{-}Lite_{\mathcal{A}}$) and
    - Query Answering (QA) over ASP
  - Implementation in DLVHEX and experiments

2. Support Sets

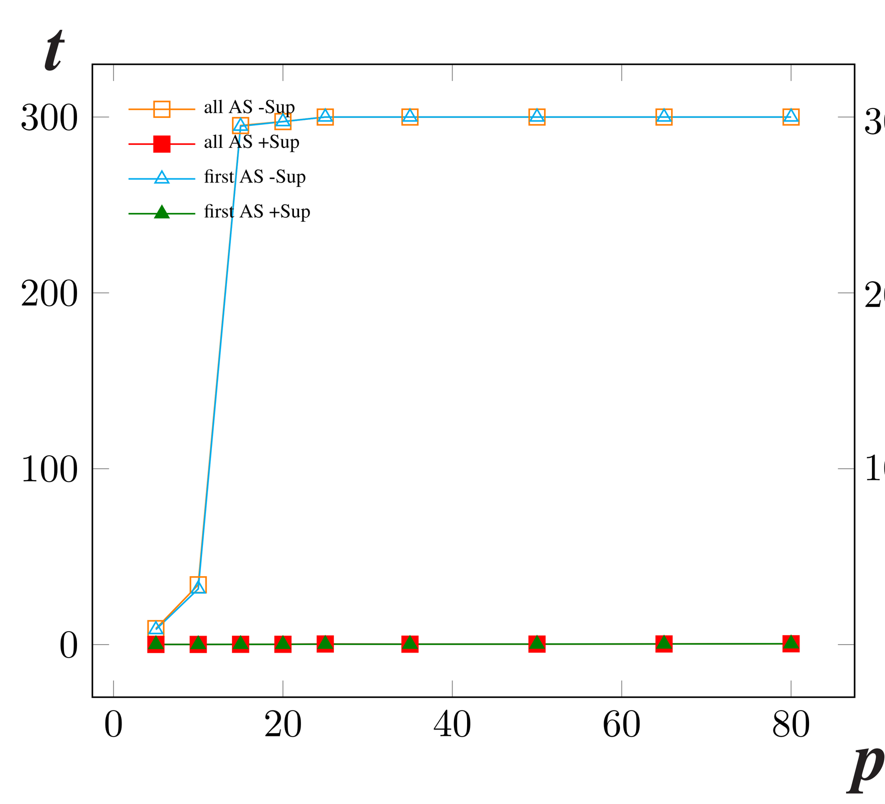
- **Support Sets** encode partial info about external source
- **Ground Support Set** for $a = \mathbf{DL}[worksIn \uplus goTo; \neg Cust](d1)$: $S = \{\mathbf{T}goTo(d_1, r4)\}$ for all assignments $\mathbf{A} \supseteq S$: $\mathbf{A} \models a$
- **Complete Support Family** $\mathcal{S}$ for a: for all $\mathbf{A}$ there is $S \in \mathcal{S}$: $\mathbf{A} \supseteq S$
- **Nonground Support Set** S for $a(\mathbf{X})$ is of form $\langle N, \gamma \rangle$, where
  - N: set of signed nonground literals over input predicates of $a(X)$
  - γ: function, selecting groundings of N, that are support sets for $a(c)$
- $S_1 = \langle \{\mathbf{T}goTo(X, X')\}, \top \rangle$, where $\top$ returns 1 for all groundings of $\mathbf{T}goTo(X, X')$

- $S_2 = \langle \emptyset, \gamma \rangle$, where $\gamma : \mathcal{C} \times \{\emptyset\} \rightarrow \{0, 1\}$ is s.t. $\gamma(c, \emptyset) = 1$ iff $EDriver(c) \in \mathcal{A}$ of $\mathcal{O}$

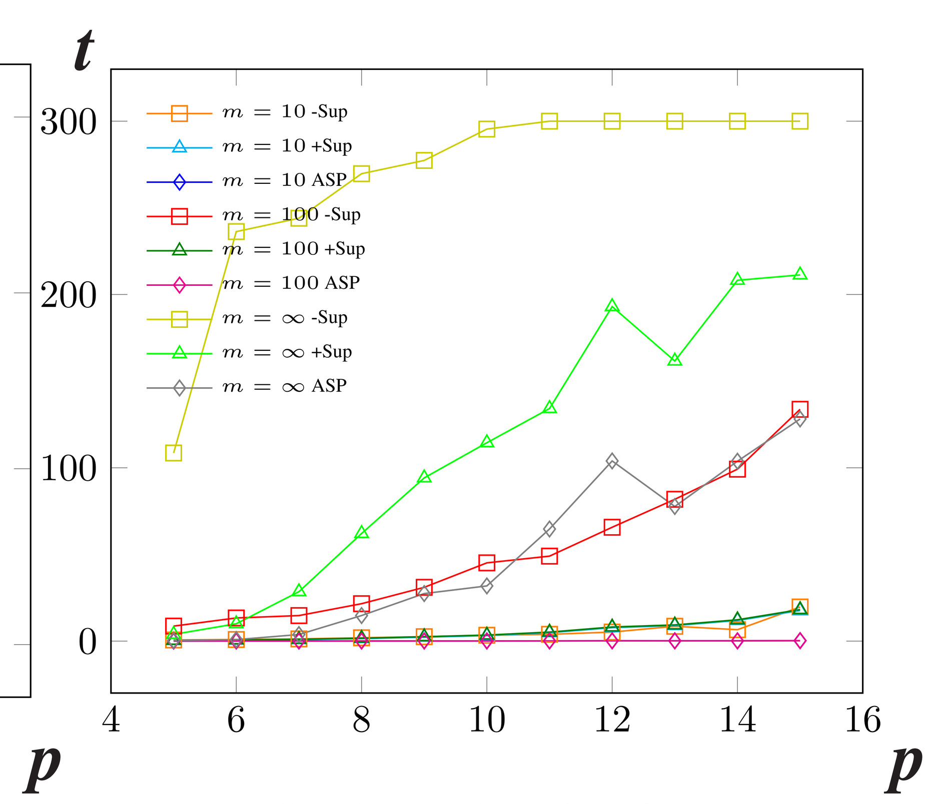
3. Using Support Sets

- Standard HEX-Program Evaluation:
  - From Π construct $\hat{\Pi}$ with all external a substituted by e_a
  - Add $e_a \vee ne_a$ to $\hat{\Pi}$, where ne_a corresponds to negation of e_a
  - For each $\hat{\mathbf{A}} \in AS(\hat{\Pi})$, check compatibility (i.e. $\mathbf{T}e_a \in \hat{\mathbf{A}}$ iff $\mathbf{A} \models a$?) and minimality (i.e. exclude self-support)
- **New Approach:**
  - **Support Sets in AS Search**: for $S \in \mathcal{S}^+(a)$ (resp. $S \in \mathcal{S}^-(a)$) adding $S \cup \{\mathbf{F}e_a\}$ (resp. $S \cup \{\mathbf{T}e_a\}$) to Π prunes not compatible $\mathbf{A}$
  - **Compatibility Check**: with complete support families for all a of Π **external source accesses can be fully eliminated**
- Support sets must be small and easily computable!
- DL-programs over consistent $DL\text{-}Lite_{\mathcal{A}}$ ontologies:
  - size is at most 2
  - computation is tractable
- Also: QA over positive ASP (e.g. subproblems)

4. Benchmark Results



Driver-Customer Assignment Problem



House Configuration Problem

- t: time in seconds, p: size of instance
- m: bound on number of computed AS
- +Sup (-Sup): using support sets (resp. standard computation)
- first (all) AS: computing first (resp. all) answer sets

Other benchmarks:

- Default rules over university LUBM ontology
- Graph non-3-colorability problem

5. Conclusion and Outlook

Results:

- Support sets are viewed as knowledge compilation
- Experimental results show significant improvements for practical applications: DL-programs over $DL\text{-}Lite_{\mathcal{A}}$ and QA over postive ASP

Future work:

- Sophisticated algorithms for nogood grounding, coverage checking
- Exploiting info about the program by support sets
- Further optimization techniques

- Calvanese, D.; Lembo, D.; Lenzerini, M.; and Rosati, R. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning*, 2007, 39(3):385–429.
- Eiter, T.; Fink, M.; Krennwallner, T.; Redl, C.; and Schüller, P. Efficient HEX-program evaluation based on unfounded sets. *Journal of Artificial Intelligence Research*, volume 49, pages 269-321, February 2014.