Eliminating Unfounded Set Checking for HEX-Programs

Thomas Eiter Michael Fink Thomas Krennwallner
Christoph Redl Peter Schüller

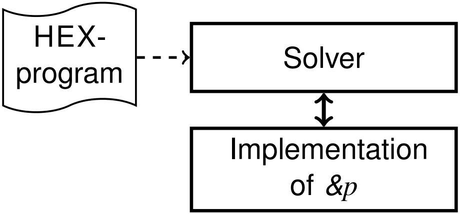
KBS Group, Institute of Information Systems, Vienna University of Technology

5th Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP 2012)
Sep 4, 2012

Motivation

HEX-programs

- extend ordinary ASP programs by external atoms *&p*
- allows to access external knowledge



Answer Set Computation

Issue: efficient computation of Answer Sets for HEX-programs

- Faber-Leone-Pfeifer (FLP) semantics [Faber *et al.*, 2011] (minimal models of FLP-reduct)
- issue of nonmonotonic external atoms with recursion
- reasoning from Horn-programs with poly external atoms is Σ_2^p-hard
- thus: answer set checking needs special care

Answer Set Computation

Issue: efficient computation of Answer Sets for HEX-programs

- simple search for smaller models does not scale
- Unfounded Sets (UFS): used to reduce FLP answer set checking to a search for UFS (implemented as a SAT problem) [Eiter *et al.*, 2012b]
- Here: find syntactic criterions to avoid UFS checking

HEX-Programs

Definition (HEX-programs)

A HEX-program consists of rules of form

$$a_1 \vee \cdots \vee a_n \leftarrow b_1, \ldots, b_m, \text{not } b_{m+1}, \ldots, \text{not } b_n,$$

with classical literals a_i, and classical literals or an external atoms b_j.

Definition (External Atoms)

An external atom is of the form

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

p ... external predicate name

q_i ... predicate names or constants

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
$f_{\&p}(\mathbf{A}, q_1, \ldots, q_k, t_1, \ldots, t_l) = 1$.

Examples

The $\&rdf$ External Atom

- Input: URL
- Output: Set of triplets from RDF file

External knowledge base is a set of RDF files on the web:

$$addr(\texttt{http://.../data1.rdf}).$$
$$addr(\texttt{http://.../data2.rdf}).$$
$$bel(X,Y) \leftarrow addr(U), \&rdf[U](X,Y,Z).$$

Examples

The $\&rdf$ External Atom

- Input: URL
- Output: Set of triplets from RDF file

External knowledge base is a set of RDF files on the web:

$$addr(\texttt{http://.../data1.rdf}).$$
$$addr(\texttt{http://.../data2.rdf}).$$
$$bel(X,Y) \leftarrow addr(U), \&rdf[U](X,Y,Z).$$

$\&diff[p,q](X)$: all elements X, which are in p but not in q:

$$\begin{array}{rcl} dom(X) & \leftarrow & \#int(X). \\ nsel(X) & \leftarrow & dom(X), \&diff[dom, sel](X). \\ sel(X) & \leftarrow & dom(X), \&diff[dom, nsel](X). \\ & \leftarrow & sel(X1), sel(X2), sel(X3), X1 \neq X2, X1 \neq X3, X2 \neq X3. \end{array}$$

Evaluation Method

Translation Approach

HEX-Program Π:

$$p(c_1).\ dom(c_1).\ dom(c_2).\ dom(c_3).$$
$$p(X) \leftarrow dom(X), \&empty[p](X).$$

Guessing program $\hat{\Pi}$:

$$p(c_1).\ dom(c_1).\ dom(c_2).\ dom(c_3).$$
$$p(X) \leftarrow dom(X), e_{\&empty[p]}(X).$$
$$e_{\&empty[p]}(X) \vee \neg e_{\&empty[p]}(X) \leftarrow dom(X).$$

8 candidates, e.g.:

$\{\mathbf{T}p(c_1), \mathbf{T}p(c_2), \mathbf{T}dom(c_1), \mathbf{T}dom(c_2), \mathbf{T}dom(c_3),$
$\mathbf{F}e_{\&empty[p]}(c_1), \mathbf{T}e_{\&empty[p]}(c_2), \mathbf{F}e_{\&empty[p]}(c_3)\}$

Compatibility check: passed $\Rightarrow$ compatible set

Minimality Criterion

Definition (FLP-Reduct [Faber *et al.*, 2011])

For an interpretation $\mathbf{A}$ over a program Π, the FLP-reduct $f\Pi^{\mathbf{A}}$ of Π wrt. $\mathbf{A}$ is the set $\{r \in \Pi \mid \mathbf{A} \models b, \text{ for all } b \in B(r)\}$ of all rules whose body is satisfied under $\mathbf{A}$.

Definition (Answer Set)

An interpretation $\mathbf{A}$ is an answer set of program Π iff it is a subset-minimal model of the FLP reduct $f\Pi^{\mathbf{A}}$.

Example

$$dom(a). \quad dom(b). \quad p(a) \leftarrow dom(a), \&g[p](a). \quad p(b) \leftarrow dom(b), \&g[p](b).$$

where $\&g$ implements the following mapping:

$\emptyset \mapsto \{b\}; \{a\} \mapsto \{a\}; \{b\} \mapsto \emptyset; \{a,b\} \mapsto \{a,b\}$

$\mathbf{A} = \{\mathbf{T}dom(a), \mathbf{T}dom(b), \mathbf{T}p(a), \mathbf{F}p(b)\}$ is a model, but not subset-minimal model of $f\Pi^{\mathbf{A}}$: $dom(a).\ dom(b).\ p(a) \leftarrow dom(a), \&g[p](a)$

Using Unfounded Sets

Definition (Unfounded Set [Faber, 2005])

A set of atoms X is an unfounded set of Π wrt. (partial) assignment $\mathbf{A}$, iff for all $a \in X$ and all $r \in \Pi$ with $a \in H(r)$, at least one of (1)–(3) holds:

1. $\mathbf{A} \not\models B(r)$
2. $\mathbf{A} \dot{\cup} \neg.X \not\models B(r)$
3. $\mathbf{A} \models h$ for some $h \in H(r) \setminus X$

(where $\mathbf{A} \dot{\cup} \neg.X = \{\mathbf{T}a \in \mathbf{A} \mid a \notin X\} \cup \{\mathbf{F}a \in \mathbf{A}\} \cup \{\mathbf{F}a \mid a \in X\}$)

Definition (Unfounded-free Assignments)

An assignment $\mathbf{A}$ is unfounded-free wrt. program Π, iff there is no unfounded set X of Π wrt. $\mathbf{A}$ such that $\mathbf{T}a \in \mathbf{A}$ for some $a \in X$.

Theorem (FLP Answer sets)

A model $\mathbf{A}$ of a program Π is is an answer set iff it is unfounded-free.

Atom Dependency Graph

Definition (Atom Dependency)

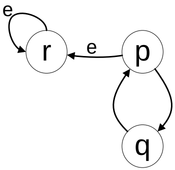
For a ground program Π and ground atoms $p(\mathbf{c})$, $q(\mathbf{d})$, we say:

(i) $p(\mathbf{c})$ depends on $q(\mathbf{d})$ ($p(\mathbf{c}) \rightarrow q(\mathbf{d})$) iff for some rule $r \in \Pi$ we have $p(\mathbf{c}) \in H(r)$ and $q(\mathbf{d}) \in B(r)$

(ii) $p(\mathbf{c})$ depends externally on $q(\mathbf{d})$ ($p(\mathbf{c}) \rightarrow_e q(\mathbf{d})$) iff for some rule $r \in \Pi$ we have $p(\mathbf{c}) \in H(r)$ and there is a $\&g[q_1, \ldots, q_n](\mathbf{d}) \in B^+(r) \cup B^-(r)$ with $q_i = q$ for some $i \in \{1, \ldots, n\}$.

Example

$\Pi = \{r \leftarrow \&id[r](); \quad p \leftarrow \&id[r](); \quad p \leftarrow q; \quad q \leftarrow p\}$



Cuts

Definition (Cut)

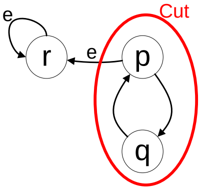
Let U be an UFS of Π wrt. $\mathbf{A}$. A set of atoms $C \subseteq U$ is a cut, if

(i) For all $a \in C, b \in U$: $b \not\to_e a$, and

(ii) For all $a \in C, b \in U \setminus C$: $b \not\to a$ and $a \not\to b$.

Lemma (Unfounded Set Reduction Lemma)

Let U be an UFS of Π wrt. $\mathbf{A}$ and let C be a cut. Then $Y = U \setminus C$ is an unfounded set of Π wrt. $\mathbf{A}$.



Example

$\Pi = \{r \leftarrow \&id[r](); \quad p \leftarrow \&id[r](); \quad p \leftarrow q; \quad q \leftarrow p\}$

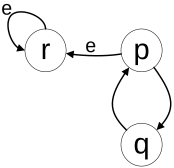
UFS $U = \{p, q, r\}$ wrt. $\mathbf{A} = \{\mathbf{T}p, \mathbf{T}q, \mathbf{T}r\}$

$\Rightarrow$ UFS $U' = \{p, q, r\} \setminus \{p, q\} = \{r\}$ wrt. $\mathbf{A}$

EA-Input Unfoundedness

Lemma (EA-Input Unfoundedness)

Let U be an unfounded set of Π wrt. $\mathbf{A}$. If there are no $x, y \in U$ s.t. $x \rightarrow_e y$, then U is an unfounded set of $\hat{\Pi}$ wrt. $\hat{\mathbf{A}}$.



Example

$\Pi = \{r \leftarrow \&id[r]();\quad p \leftarrow \&id[r]();\quad p \leftarrow q;\quad q \leftarrow p\}$

UFS $U_1 = \{p, q\}$ wrt. $\mathbf{A}' = \{\mathbf{T}p, \mathbf{T}q, \mathbf{F}r\}$ is already detected when $\hat{\Pi} = \{e_{\&id[r]}() \vee \neg e_{\&id[r]}() \leftarrow;\quad r \leftarrow e_{\&id[r]}();\quad p \leftarrow e_{\&id[r]}();\quad p \leftarrow q;\quad q \leftarrow p\}$ is evaluated

UFS $U_2 = \{p, q, r\}$ wrt. $\mathbf{A}'' = \{\mathbf{T}p, \mathbf{T}q, \mathbf{T}r\}$ is not detected during model generation phase of the ordinary part as $p, r \in U_2$ and $p \rightarrow_e r$

E-Cycles

Definition (Cycle and E-Cycle)

A cycle under a binary relation $\circ$ is a sequence of elements $C = c_0, \ldots, c_{n+1}$ $(n \geq 0)$ s.t. $(c_i, c_{i+1}) \in \circ$ for all $i \in \{0, \ldots, n\}$ and $c_0 = c_{n+1}$.

Let $\rightarrow^d = \rightarrow \cup \leftarrow \cup \rightarrow_e$ ($\leftarrow$ is the inverse of $\rightarrow$).
A cycle $c_0, \ldots, c_{n+1}$ in $\rightarrow^d$ is called an e-cycle, iff it contains e-edges.

Proposition (Relevance of e-cycles)

Suppose U is an unfounded set of Π wrt. $\mathbf{A}$ which contains no e-cycle under $\rightarrow^d$. Then there exists an unfounded set of $\hat{\Pi}$ wrt. $\hat{\mathbf{A}}$.

Corollary

If there is no e-cycle under $\rightarrow^d$ and $\hat{\Pi}$ has no unfounded set wrt. $\hat{\mathbf{A}}$, then $\mathbf{A}$ is unfounded-free for Π.

E-Cycles

Example (Programs without E-Cycles)

$\Pi_1 = \{out(X) \leftarrow \&diff[set_1, set_2](X)\} \cup F$ ($F \ldots$ set of facts)

$\Pi_2 = \{str(Z) \leftarrow dom(Z), str(X), str(Y), \text{not } \&concat[X, Y](Z)\}$

E-Cycles

Example (Programs without E-Cycles)

$\Pi_1 = \{out(X) \leftarrow \&diff[set_1, set_2](X)\} \cup F \quad (F \ldots$ set of facts)

$\Pi_2 = \{str(Z) \leftarrow dom(Z), str(X), str(Y), \text{not } \&concat[X, Y](Z)\}$

Proposition (Unfoundedness of Cyclic Input Atoms)

If U is an unfounded set of Π wrt. $\mathbf{A}$ and U contains no cyclic input atoms, then $\hat{\Pi}$ has an unfounded set wrt. $\hat{\mathbf{A}}$.

Program Decomposition

Let $\mathcal{C}$ be a partitioning of the ordinary atoms $A(\Pi)$ of Π into $\subseteq$-maximal strongly connected components under $\rightarrow \cup \rightarrow_e$.

Definition (Associated Programs)

For each $C \in \mathcal{C}$, the program associated with C is defined as

$$\Pi_C = \{r \in \Pi \mid H(r) \cap C \neq \emptyset\} \ .$$

Proposition

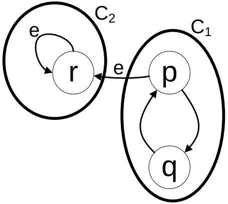
Let U be a nonempty unfounded set of Π wrt. $\mathbf{A}$. Then for some Π_C with $C \in \mathcal{C}$ we have that $U \cap C$ is an unfounded set of Π_C wrt. $\mathbf{A}$.

Proposition

Let U be a nonempty unfounded set of Π_C wrt. $\mathbf{A}$ such that $U \subseteq C$. Then U is an unfounded set of Π wrt. $\mathbf{A}$.

Program Decomposition Example

$$\Pi = \{r \leftarrow \&id[r](); \quad p \leftarrow \&id[r](); \quad p \leftarrow q; \quad q \leftarrow p\}$$



$\mathcal{C} = \{C_1, C_2\}$ with $C_1 = \{p, q\}$ and $C_2 = \{r\}$
$\Pi_{C_1} = \{p \leftarrow \&id[r](); p \leftarrow q; q \leftarrow p\}$
$\Pi_{C_2} = \{r \leftarrow \&id[r]()\}$.

Let $U = \{p, q, r\}$ be an UFS wrt. $\mathbf{A} = \{\mathbf{T}p, \mathbf{T}q, \mathbf{T}r\}$
Then $U \cap \{r\} = \{r\}$ is also an unfounded set of Π_{C_2} wrt. $\mathbf{A}$

Experiments

Implementation: dlvhex [Eiter *et al.*, 2012a]

Argumentation

#args	first answer set					all answer sets				
	standard approach		new approach			standard approach		new approach		
	timeouts	avg	timeouts	avg	gain	timeouts	avg	timeouts	avg	gain
5	0	1,09	0	1,07	2,21%	0	1,70	0	1,56	9,21%
6	0	2,40	0	2,30	4,58%	0	4,58	0	3,74	22,58%
7	0	5,58	0	5,33	4,68%	0	15,66	0	11,28	38,78%
8	0	14,26	0	12,74	11,99%	3	71,06	2	39,32	80,71%
9	0	39,82	0	33,57	18,63%	16	174,99	8	106,34	64,55%
10	2	126,54	0	80,00	58,18%	40	278,98	16	214,81	29,87%

Conclusion

Conclusion

- Decision criterion for avoiding the UFS check
- Based on concept of e-cycles
- Modular application via program decomposition
- Benchmark results are promising

Future Work

- Other syntactic criteria
- Use semantic information

References I

- Mario Alviano, Francesco Calimeri, Wolfgang Faber, Simona Perri, and Nicola Leone.

  Unfounded sets and well-founded semantics of answer set programs with aggregates.

  Journal of Artificial Intelligence Research, 42:487–527, 2011.

- Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits.

  A uniform integration of higher-order reasoning and external evaluations in answer-set programming.

  In *19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 90–96. Professional Book, 2005.

References II

- Thomas Eiter, Michael Fink, Thomas Krennwallner, and Christoph Redl.
  Conflict-driven ASP Solving with External Sources.
  Theory and Practice of Logic Programming: Special Issue 28th International Conference on Logic Programming (ICLP 2012), September 2012.

- Thomas Eiter, Michael Fink, Thomas Krennwallner, Christoph Redl, and Peter Schüller.
  Exploiting Unfounded Sets for HEX-Program Evaluation.
  In Luis Fariñas del Cerro, Andreas Herzig, and Jérôme Mengin, editors, *13th European Conference on Logics in Artificial Intelligence (JELIA 2012), September 26-28, 2012, Toulouse, France*, volume 7519 of *LNCS*. Springer, September 2012.

References III

- Wolfgang Faber, Nicola Leone, and Gerald Pfeifer.

  Semantics and complexity of recursive aggregates in answer set programming.

  Artif. Intell., 175(1):278–298, 2011.

- Wolfgang Faber.

  Unfounded sets for disjunctive logic programs with arbitrary aggregates.

  In *8th International Conference Logic Programming and Nonmonotonic Reasoning (LPNMR'05)*, volume 3662, pages 40–52. Springer, 2005.