∃-ASP for computing repairs with existential ontologies

Jean-François Baget (1)  Zied Bouraoui (2)  Farid Nouioua (3)
Odile Papini (3)  Swan Rocher (4)  Eric Würbel (3)

(1) INRIA, France. (2) Cardiff University, UK. (3) Aix-Marseille University, France. (4) Montpellier University, France.

ASPIQ
ASP technologIes for Querying large scale multisource heterogeneous web information
The aim

Framework : Existential rules

Inconsistency-tolerant inference
The aim

Framework: Existential rules

- Inconsistency-tolerant inference
- Repairs

Unified framework

[Baget, Benferhat, Bouraoui, Croitoru, Mugnier, Papini, Rocher, Tabia, KR 2016]
The aim

Framework: Existential rules

Inconsistency-tolerant inference

∃-ASP Program

∃-ASP [Garreau, Garcia, Lefèvre, Stéphan, ONTOLP 2015]
The aim

Framework: Existential rules

- Inconsistency-tolerant inference
- ∃-ASP Program
- ASPeRiX
- Repairs
- Answer Sets
Overview

1. Preliminaries
   - Existential rules
   - $\exists$-ASP

2. From inconsistency-tolerant inferences to $\exists$-ASP

3. One-to-one correspondence between answer sets and repairs

4. Conclusion
Existential rules

Knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$

$\mathcal{F}$: Set of facts
Existential closure of conjunction of atoms

$\mathcal{R}$: Set of existential rules
rules of the form $B \rightarrow H$ where body $B$ and head $H$ are conjunction of atoms.
$B[\vec{X}, \vec{Y}] \rightarrow H[\vec{Y}, \vec{Z}]$  \hspace{1cm} $\vec{X}, \vec{Y}$: universally quantified, $\vec{Z}$: existential variable
FOL: $\forall \vec{X}, \forall \vec{Y} (\phi(B) \rightarrow (\exists \vec{Z} \phi(H)))$.

$\mathcal{N}$: Set of constraints
rules of the form $B \rightarrow \bot$ where $B$ is a set of atoms

Inconsistent Knowledge base $K$ : \hspace{0.5cm} $\bot \in \text{Cl}(\mathcal{F}, \mathcal{R} \cup \mathcal{N})$
Existential rules

Classes where the skolem chase stops [Baget, Garreau, Mugnier, Rocher. NM2014]
Repairs

\[ \mathcal{K} = (F, \mathcal{R}, \mathcal{N}) \] with \( F \) : set of ground atoms

**standard repair of** \( \mathcal{K} : R(\mathcal{K}) \)

An Inclusion-maximal subset of \( F \) (consistent w.r.t. \((\mathcal{R}, \mathcal{N})\))

\[ g^+ Cl(X) : \] ground positive closure of a set of atoms \( X \). The restriction of \( Cl(X, \mathcal{R}) \) to basic ground atoms

**closed repair of** \( \mathcal{K} : CR(\mathcal{K}) \)

A set of basic ground atoms \( g^+ Cl(F') \), where \( F' \) is a standard repair of \( \mathcal{K} \).

**repairs of the closure of** \( \mathcal{K} : RC(\mathcal{K}) \)

A standard repair of \((g^+ Cl(F, \mathcal{R}), \mathcal{R}, \mathcal{N})\).
∃-ASP

∃-ASP Program

Π a set of rules of the form:

\[ H \leftarrow B^+, \text{not } N_1^-, \ldots, \text{not } N_k^- . \]

head \( H \), positive body \( B^+ \), negative bodies \( N_i^- \): sets of basic atoms

Skolemization of an ∃-ASP Program

\[ \Pi = \Pi_F \cup \Pi_R \quad \Pi_F : \text{basic atoms,} \quad \Pi_R : \exists\text{-ASP rules} \]

Skolemization of \( \Pi \): grounding of \( \Pi_F \) + skolemization of \( \Pi_R \)

Answer set

An answer set of ∃-ASP program: an answer set of its skolemization
Answer sets computing

The Computation Tree (see ASPeRiX)

- Let $F$ be a set of facts and $R$ be a set of $\exists$-ASP rules

Let $R \in R$ s.t. $\sigma(B) \in IN$ and there is no $i$ s.t. $\sigma(N_i) \in IN$

Repeat for each obtained node by applying existential rules breadth-first.
The Computation Tree (cont)

- The computation tree generates a (possibly infinite) tree.
- It is **complete** when no further rule application can add new atoms in IN.
- The **result** of a complete branch is the union of all IN found in that branch.
- A branch is **OUT-valid** when no fact present in a OUT of that branch is also present in the result.
- A branch is **MBT-valid** when all facts present in a MBT of that branch are also present in the result.

An *answer set* of \((F, \cdot R)\) is the result of a complete, OUT and MBT valid branch of a computation tree from \((F, \cdot R)\).
From inconsistency-tolerant inferences to $\exists$-ASP

Transformation into ASP

Overall Framework

According to the selection and display rules provided, the restriction of any answer set $\text{AS}_i$ of $\Pi$ to the original vocabulary will be some repair of $(F, R, N)$.
Transformation into ASP

From inconsistency-tolerant inferences to $\exists$-ASP

Inconsistency-tolerant inference

ASP Program

from $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ with vocabulary $\mathcal{V}$

- Skolemization: $(\mathcal{F}, \mathcal{R}^{sk}, \mathcal{N})$

- Extension of the vocabulary $\mathcal{V} \rightarrow \mathcal{V'}$:

  For any $p \in \mathcal{V}$
  - $p_i$: initial
  - $p_s$: may be selected
  - $p_v$: valid
  - $p_p$: possible
  - $p_c$: chosen
  - $p_g$: ground
  - $p_n$: forbidden
  - $p_d$: display
Transformation into ASP

For every $p(\vec{t}) \in F$, $p_i(\vec{t})$ : initial predicate
From inconsistency-tolerant inferences to $\exists$-ASP

Transformation into ASP

- For every $p(\vec{t}) \in F$ \quad $p_i(\vec{t})$ : initial predicate
- For every predicate in $\forall : [P_1 :] \; p_p(\vec{X}) \leftarrow p_i(\vec{X})$
For every $p(t) \in F$, $p_i(t)$: initial predicate
For every predicate in $\mathcal{V}$, $[P_1 : p_p(\vec{X}) \leftarrow p_i(\vec{X})$.
For every rule in $\mathcal{R}_{sk}$, $[R_1 : H_p(\vec{X}, \vec{Y}), fct(Y_1), \cdots, fct(Y_k) \leftarrow B_p(\vec{X})$. 
From inconsistency-tolerant inferences to $\exists$-ASP

Transformation into ASP

Computation tree (1)

- For every $p(\vec{t}) \in F$, $p_i(\vec{t})$: initial predicate
- For every predicate in $\mathcal{V}$, $[P_1 :] p_p(\vec{X}) \leftarrow p_i(\vec{X})$.
- For every rule in $\mathcal{R}_{sk}$, $[R_1 :] H_p(\vec{X}, \vec{Y}), fct(Y_1), \cdots, fct(Y_k) \leftarrow B_p(\vec{X})$.
- For every predicate in $\mathcal{V}$, $[P_2 :] p_g(\vec{X}) \leftarrow p_p(\vec{X}), notfct(X_1), \cdots notfct(X_k)$. 
A selection strategy selects some finite subset $X$ of atoms and marks them “s”.

For instance:
- Select all initial atoms
- Select all ground atoms
From inconsistency-tolerant inferences to $\exists$-ASP

Transformation into ASP

Computation tree (3) : Choice

$[P_3 :] p_c(\vec{X}) \leftarrow p_s(\vec{X}), \text{not } p_n(\vec{X})$
From inconsistency-tolerant inferences to $\exists$-ASP

Transformation into ASP

Computation tree (3) : Choice

\[ P_3 : p_c(\vec{X}) \leftarrow p_s(\vec{X}), \text{not } p_n (\vec{X}) \]
From inconsistency-tolerant inferences to \( \exists \)-ASP

Transformation into ASP

Computation tree (3) : Choice

Jean-François Baget (1), Zied Bouraoui (2), Farid Nouioua (3), Odile Papini (3), Swan Rocher (4), Eric Würbel (3) (1) INRIA, France. (2) Cardiff University, UK. (3) Aix-Marseille University, France. (4) Montpellier University, France.
Transformation into ASP

Computation tree (3) : Choice

For every subset \( Y \) of \( X \), there is a branch where we mark every atom of \( Y \) as chosen \((c)\). The leaf has the following form:
From inconsistency-tolerant inferences to $\exists$-ASP

Transformation into ASP

Computation tree (4) : Contexts

$[P_4 :] p_v(\bar{X}, \text{base}) \leftarrow p_c(\bar{X})$. 
Computation tree (4) : Contexts

\[ P_4 : ] p_v(\vec{X}, base) \leftarrow p_c(\vec{X}). \]
\[ P_5 : ] p_v(\vec{X}, ctx(p, \vec{X})), context(ctx(p, \vec{X})) \leftarrow p_s(\vec{X}), not p_c(\vec{X}). \]
[\text{Computation tree (4) : Contexts}]

\[ P_4 : p_v(\vec{X}, \text{base}) \leftarrow p_c(\vec{X}). \]
\[ P_5 : p_v(\vec{X}, \text{ctx}(p, \vec{X})), \text{context}((\text{ctx}(p, \vec{X}))) \leftarrow p_s(\vec{X}), \text{not } p_c(\vec{X}). \]
\[ P_6 : p_v(\vec{X}, C) \leftarrow p_v(\vec{X}, \text{base}), \text{context}(C). \]
Transformation into ASP

Computation tree (5) : Propagation

For every rule in $\mathcal{R}_{sk}$, $B(\bar{X}) \rightarrow H(\bar{X})$, $[R_2 : ] H_v(\bar{X}, C) \leftarrow B_v(\bar{X}, C)$. 

Jean-François Baget (1), Zied Bouraoui (2), Farid Nouioua (3), Odile Papini (3), Swan Rocher (4), Eric Würl (3) (1) INRIA, France. (2) Cardiff University, UK. (3) Aix-Marseille University, France. (4) Montpellier University, France.)
From inconsistency-tolerant inferences to ∃-ASP

Transformation into ASP

Computation tree (5) : Propagation

For every rule in $R_{sk}$, $B(\vec{X}) \rightarrow H(\vec{X})$, $[R_2 :]$ $H_v(\vec{X}, C) \leftarrow B_v(\vec{X}, C)$.

For every constraint in $N$ $[C_1 :]$, $\text{absurd}(C) \leftarrow p_1^v(\vec{X}_1, C), \ldots, p_k^v(\vec{X}_k, C)$.

- $Y_c$ is consistent iff $\text{absurd}(\text{base})$ is not derived
- $Y_c \cup \{a\}$ is consistent iff $\text{absurd}(\text{ctx}(a))$ is not derived

Finally

- $Y_c$ is a maximal consistent subset of $X_s$ when $\text{absurd}(\text{base})$ is not derived and for every other context $\text{ctx}(a)$ $\text{absurd}(\text{ctx}(a))$ is derived

Jean-François Baget (1), Zied Bouraoui (2), Farid Nouioua (3), Odile Papini (3), Swan Rocher (4), Eric Würgel (3) (1) INRIA, France. (2) Cardiff University, UK. (3) Aix-Marseille University, France. (4) Montpellier University, France.)

∃-ASP for computing repairs with existential ontologies
Transformation into ASP

Computation tree (6) : Evaluation

Case 1 : absurd base context

\[ [C_2 : \neg p_n(\overline{X}) \leftarrow p_c(\overline{X}), \text{absurd}(\text{base})]. \]

\( Y_c \) is not consistent and the branch is not OUT-valid
Transformation into ASP

Computation tree (6) : Evaluation

Case 2 : base context not absurd, but all the other contexts are absurd

\([C_3 \; [: p_n(\vec{X}) \leftarrow p_s(\vec{X}), \; \text{not} \; p_c(\vec{X}), \; p_v(\vec{X}, C), \; \text{context}(C), \; \text{absurd}(C)].\)

\(Y_c\) is consistent and the branch is OUT-valid and MBT-valid
Computation tree (6) : Evaluation

**Case 3** : One non-base context is not absurd

\[ C_3 : p_n(\bar{X}) \leftarrow p_s(\bar{X}), \text{not} \ p_c(\bar{X}), \ p_v(\bar{X}, C), \text{context}(C), \text{absurd}(C). \]

\( Y_c \) is not maximal consistent and the branch is not MBT-valid
Transformation into ASP

Computation tree (7) : Display

Display strategy: Selects some finite subset of atoms valid in the base context and restricts them to the original vocabulary

- all initial atoms in the base context
- all ground atoms in the base context
One-to-one correspondence between answer sets and repairs

Unified framework Semantics

- $\circ_1$ or $R$ computes the set of repairs of $\mathcal{K} : R(\mathcal{K})$
- $\circ_5$ or $CR$ computes the closed repairs of $\mathcal{K} : CR(\mathcal{K})$
- $\circ_7$ or $RC$ computes the repairs of the closure of $\mathcal{K} : RC(\mathcal{K})$

SEL1 / DISP1 : Select / Display all atoms
SEL2 / DISP2 : Select / Display all ground atoms

One-to-one correspondence

Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ a knowledge base.
$\rho(AS)$ denotes the restriction of $AS$ to the original vocabulary $\mathcal{V}$

<table>
<thead>
<tr>
<th>$\exists$-ASP program</th>
<th>SEL</th>
<th>DISP</th>
<th>$\rho(AS(\Pi))$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Pi_1$</td>
<td>SEL1</td>
<td>DISP1</td>
<td>$\rho(AS(\Pi_1))$</td>
<td>repairs of $\mathcal{K} : R(\mathcal{K})$</td>
</tr>
<tr>
<td>$\Pi_5$</td>
<td>SEL1</td>
<td>DISP2</td>
<td>$\rho(AS(\Pi_5))$</td>
<td>closed repairs of $\mathcal{K} : CR(\mathcal{K})$</td>
</tr>
<tr>
<td>$\Pi_7$</td>
<td>SEL2</td>
<td>DISP2</td>
<td>$\rho(AS(\Pi_7))$</td>
<td>repairs of the closure of $\mathcal{K} : RC(\mathcal{K})$</td>
</tr>
</tbody>
</table>

Jean-François Baget (1), Zied Bouraoui (2), Farid Nouioua (3), Odile Papini (3), Swan Rocher (4), Eric Würl (3)
(1) INRIA, France. (2) Cardiff University, UK. (3) Aix-Marseille University, France. (4) Montpellier University, France.

∃-ASP for computing repairs with existential ontologies
Conclusion

- generic encoding in $\exists$-ASP of repair-based techniques
- allows for computing the semantics: AR, IAR, ICR, $\langle \diamond_7, \forall \rangle$, $\langle \diamond_7, \cap \rangle$

Let $q$: a query, $q_v$ obtained from $q$ by replacing any $p(t)$ by $p_v(t, \text{base})$

- $\mathsf{AR}$ $\mathcal{K} \models_{\langle \diamond_1, \forall \rangle} q$ iff $\forall AS \in AS(\Pi_1)$, $q_v \in AS$.
- $\mathsf{IAR}$ $\mathcal{K} \models_{\langle \diamond_1, \cap \rangle} q$ iff $q_v \in \bigcap_{AS_i \in AS(\Pi_1)} AS_i$.
- $\mathsf{ICR}$ $\mathcal{K} \models_{\langle \diamond_5, \cap \rangle} q$ iff $q_v \in \bigcap_{AS_i \in AS(\Pi_5)} AS_i$.
- $\langle \diamond_7, \forall \rangle$ $\mathcal{K} \models_{\langle \diamond_7, \forall \rangle} q$ iff $\forall AS \in AS(\Pi_7)$, $q_v \in AS$. (Closed to $\mathsf{CAR}$)
- $\langle \diamond_7, \cap \rangle$ $\mathcal{K} \models_{\langle \diamond_7, \cap \rangle} q$ iff $q_v \in \bigcap_{AS_i \in AS(\Pi_7)} AS_i$. (Closed to $\mathsf{ICAR}$)

- future works:
  - implementation and experimentation
  - extension to repairs minimal w.r.t. cardinality