Inconsistencies in Hybrid Knowledge Bases

Daria Stepanova
Faculty Mentors: Thomas Eiter and Michael Fink
Vienna University of Technology, Austria

1. Motivation

Hybrid Knowledge Bases: combination of different logical formalisms

- Loose coupling (DL-programs, F-Logic,..)
- Tight coupling (SWRL, ELN,..)
- Embedding (MKNF, Open ASP,..)

Inconsistencies easily emerge in HKBs

DL-program \( \Pi = (O, P) \) is inconsistent

\[ O = \{(1) \text{Child} \sqsubseteq \exists \text{hasParent}, (4) \text{Male}(\text{pat}), (3) \text{Female} \sqsubseteq \neg \text{Male} \} \]

\[ P = \{(7) \text{ischildof}(\text{john}, \text{alex}); (8) \text{boy}(\text{john}); (9) \text{hasfather}(\text{john}, \text{pat}) \sqsubseteq \text{DL:Male} \sqcap \text{boy}; \text{Male}(\text{pat}), \text{DL}; \text{hasParent}(\text{john}, \text{pat}); (10) \neg \text{not DL:Adopted}(\text{john}, \text{pat} \neq \text{alex}) \}
\]

\[ \mathcal{A} = \{ \text{Male}(\text{john}), \exists \text{hasParent}(\text{john}, \text{pat}) \} \]

- Aim of this work: methodology for repairing Hybrid KBs (at \( O \) side)

- Contributions:
  - Framework for repair computation and its complexity
  - Implementation and evaluation of developed framework

2. DL-programs

- **DL-program**: ontology + rules (loose-coupling approach)
- **DL-atoms** serve as query interfaces to ontology
- **Bidirectional information flow between ontology and rules**

\[ \Pi = (O, P) \text{ is a DL-program}
\]

\[ \mathcal{O} = \{(1) \text{Child} \sqsubseteq \exists \text{hasParent}, (4) \text{Male}(\text{pat}), (3) \text{Female} \sqsubseteq \neg \text{Male} \}
\]

\[ \mathcal{P} = \{(7) \text{ischildof}(\text{john}, \text{alex}); (8) \text{boy}(\text{john}); (9) \text{hasfather}(\text{john}, \text{pat}) \sqsubseteq \text{DL:Male} \sqcap \text{boy}; \text{Male}(\text{pat}), \text{DL}; \text{hasParent}(\text{john}, \text{pat}); (10) \neg \text{not DL:Adopted}(\text{john}, \text{pat} \neq \text{alex}) \}
\]

3. DL-program Evaluation

Given:

\[ \Pi = (O, P), P \{ (r) \subset consequent \sqsubseteq \text{DL:con}\} \]

\[ \mathcal{O} = \{C \sqsubseteq D; A(c)\}
\]

Construct:

\[ \Pi \{ (r) : \text{con} \sqsubseteq \text{DL:con}\}
\]

Compute:

Answer sets of \( \Pi \):

\[ \text{AS}(\Pi) = \{1\}_{\text{r}, \text{con}, \text{q}} \]

Check:

- Compatibility: \( \Pi \{ (r) : \text{con} \sqsubseteq \text{DL:con}\} \]
- Minimality: \( \Pi \{ (r) : \text{con} \sqsubseteq \text{DL:con}\} \)

Reasons for Inconsistency:

- Repair semantics for DL-programs and its complexity (IJCAI’13)
- Independent repair selection functions
- Sound and complete deletion repair algorithm
- Support sets as optimization means (AAAI’14)
- Usage of complete support families for DL-life (ECAI’14, DL’14)
- Usage of incomplete support families for EL (ECAI’14)
- Implementation within divhlex framework, evaluation
- Independent DL-atoms, calculus for their derivation (RR’12)

4. Repair Approach

Given:

\[ \Pi = (O, P), \text{s.t. } P \{ (r) \subset consequent \sqsubseteq \text{DL:con}\}
\]

\[ \mathcal{O} = \{E \sqsubseteq D; A \sqsubseteq D; A(c); \neg C(c); E\}
\]

Compute support sets for \( a_1(X) \):

\[ S_{a_1} = \{ \{D(X)\}, \{A(X)\}, \{E(X)\}, \{\neg C(Y), C(Y)\} \}
\]

\[ S_{a_2} = \{ \{\neg C(X)\}, \{D(Y)\}, \{\neg D(Y)\}, \{\neg E(Y), E(Y)\} \}
\]

For each \( i \in \text{AS}(\Pi) \):

\[ I = \{p(X), r(X), q(X), e_1\} \]

Construct Ontology Repair Problem (ORP) \( P = (O, D_1, D_2) \), where

\[ D_1 = \{\{\neg C(X)\}, D(C)\}, D_2 = \{\{D(C), \neg E(C)\}, \neg C(C)\}
\]

Ground support sets \( S_a \):

\[ \text{Grnd}(S_{a_1}, I, A) = \{\{A(E)\}, \{E\}\}
\]

\[ \text{Grnd}(S_{a_2}, I, A) = \{\{\neg C(E)\}, \{E, E\}\}
\]

Compute Repair \( A' \) for \( P \) s.t.

\[ Q' = (T', A') \text{ is consistent, } Q' \cup \{\neg C(c)\} \sqsubseteq D(C), \neg C\}

\[ Q' \cup \{D(C), \neg E(C)\} \neq \neg C(C)
\]

\[ A' = \{A(c), \neg C(c), E(c)\} \text{ is a deletion repair!}
\]

5. Results

- Further benchmark construction and evaluation
- Size bounded and other preferred repairs (implementation)
- Complements conditions on support families for EL
- Wrapping up...

6. Future Work

- Development of bidirectional information flow between ontology and rules
- Further research on repair algorithms
- Development of repair systems for real-world applications