Towards Practical Deletion Repair of Inconsistent DL-programs

Thomas Eiter   Michael Fink   Daria Stepanova

Knowledge-Based Systems Group,
Institute of Information Systems,
Vienna University of Technology
http://www.kr.tuwien.ac.at/

ECAI 2014 –August, 21, 2014
Motivation

- **DL-program**: consistent ontology $\mathcal{O}$ + rules $\mathcal{P}$ (loose coupling combination approach)
- DL-atoms serve as query interfaces to $\mathcal{O}$
- Possibility to add information from $\mathcal{P}$ to $\mathcal{O}$ prior to querying it allows for bidirectional information flow
Motivation

- **DL-program**: consistent ontology $\mathcal{O} +$ rules $\mathcal{P}$ (loose coupling combination approach)
- DL-atoms serve as query interfaces to $\mathcal{O}$
- Possibility to add information from $\mathcal{P}$ to $\mathcal{O}$ prior to querying it allows for bidirectional information flow

However, information exchange between $\mathcal{P}$ and $\mathcal{O}$ can cause **inconsistency** of the DL-program (absence of answer sets).

[!Eiter et al, *IJCAI’2013*] Repair answer sets and algorithm for repairing ontology data part, but the latter **lacks practicality**.
Motivation

- **DL-program**: consistent ontology $O +$ rules $P$ (loose coupling combination approach)
- DL-atoms serve as query interfaces to $O$
- Possibility to add information from $P$ to $O$ prior to querying it allows for bidirectional information flow

However, information exchange between $P$ and $O$ can cause **inconsistency** of the DL-program (absence of answer sets).

[Eiter et al, *IJCAI’2013*] Repair answer sets and algorithm for repairing ontology data part, but the latter **lacks practicality**.

**In this work**: Algorithm for DL-program repair based on support sets for DL-atoms. Effective for ontologies in $DL$-$Lite_A$. 
Overview

Motivation

DL-programs

Support Sets for DL-atoms

Repair Answer Set Computation

Experiments

Conclusion
**DL-Lite**

- Lightweight Description Logic for accessing large data sources
- Concepts and roles model sets of objects and their relationships

\[ C \rightarrow A \mid \exists R \quad R \rightarrow P \| P^- \]

- A *DL-Lite* ontology \( O = \langle T, A \rangle \) consists of:
  - **TBox** \( T \) specifying constraints at the conceptual level
    \[ C_1 \sqsubseteq C_2, \quad C_1 \sqsubseteq \neg C_2, \]
    \[ R_1 \sqsubseteq R_2, \quad R_1 \sqsubseteq \neg R_2, \quad (\text{funct } R) \]
  - **ABox** \( A \) specifying the facts that hold in the domain
    \[ A(b) \quad P(a, b) \]
DL-Lite$_A$

- Lightweight Description Logic for accessing large data sources
- Concepts and roles model sets of objects and their relationships

\[ C \rightarrow A \mid \exists R \quad R \rightarrow P \mid P^- \]

- A DL-Lite$_A$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ consists of:
  - **TBox** $\mathcal{T}$ specifying constraints at the conceptual level
    \[ C_1 \sqsubseteq C_2, \quad C_1 \sqsubseteq \neg C_2, \quad R_1 \sqsubseteq R_2, \quad R_1 \sqsubseteq \neg R_2, \quad (\text{funct } R) \]
  - **ABox** $\mathcal{A}$ specifying the facts that hold in the domain
    \[ A(b) \quad P(a, b) \]

Example

\[ \mathcal{T} = \left\{ \begin{array}{c}
\text{Child} \sqsubseteq \exists \text{hasParent} \\
\text{Female} \sqsubseteq \neg \text{Male}
\end{array} \right\} \quad \mathcal{A} = \left\{ \begin{array}{c}
\text{hasParent}(\text{john, pat}) \\
\text{Male} (\text{john})
\end{array} \right\} \]
**DL-Lite\(\mathcal{A}\)**

- Lightweight Description Logic for accessing large data sources
- Concepts and roles model sets of objects and their relationships

\[
C \rightarrow A \mid \exists R \quad R \rightarrow P|P^-
\]

- A **DL-Lite\(\mathcal{A}\)** ontology \(\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle\) consists of:
  - **TBox** \(\mathcal{T}\) specifying constraints at the conceptual level
    \[
    C_1 \sqsubseteq C_2, \quad C_1 \sqsubseteq \neg C_2, \\
    R_1 \sqsubseteq R_2, \quad R_1 \sqsubseteq \neg R_2, \quad (\text{funct } R)
    \]
  - **ABox** \(\mathcal{A}\) specifying the facts that hold in the domain
    \[
    A(b) \quad P(a, b)
    \]

- For query derivation: **single** ABox assertion
- For inconsistency: at most **two** ABox assertions
- Classification is **tractable**

[Calvanese et al., 2007]
Example: DL-program

\[ \Pi = \langle O, P \rangle \] is a DL-program

\[ O = \{ (1) \text{Child} \sqsubseteq \exists \text{hasParent} \quad (4) \text{Male}(\text{pat}) \\
(2) \text{Adopted} \sqsubseteq \text{Child} \quad (5) \text{Male}(\text{john}) \\
(3) \text{Female} \sqsubseteq \neg \text{Male} \quad (6) \text{hasParent}(\text{john, pat}) \} \]
Example: DL-program

\[ \Pi = \langle \mathcal{O}, \mathcal{P} \rangle \] is a DL-program

\[ \mathcal{O} = \left\{ \begin{array}{ll}
(1) & \text{Child} \sqsubseteq \exists \text{hasParent} \\
(2) & \text{Adopted} \sqsubseteq \text{Child} \\
(3) & \text{Female} \sqsubseteq \neg \text{Male}
\end{array} \right\} \]

\[ \mathcal{P} = \left\{ \begin{array}{ll}
(4) & \text{Male}(\text{pat}) \\
(5) & \text{Male}(\text{john}) \\
(6) & \text{hasParent}(\text{john}, \text{pat}) \\
(7) & \text{ischildof}(\text{john}, \text{alex}); \\
(8) & \text{boy}(\text{john}); \\
(9) & \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[\text{Male} \uplus \text{boy}; \text{Male}](\text{pat}), \\
& \text{DL}[, \text{hasParent}](\text{john}, \text{pat})
\end{array} \right\} \]
Example: DL-program

$\Pi = \langle \mathcal{O}, \mathcal{P} \rangle$ is a DL-program

$\mathcal{O} = \{ (1) \text{Child} \sqsubseteq \exists \text{hasParent} \quad (4) \text{Male}(\text{pat}) \\
(2) \text{Adopted} \sqsubseteq \text{Child} \quad (5) \text{Male}(\text{john}) \\
(3) \text{Female} \sqsubseteq \neg \text{Male} \quad (6) \text{hasParent}(\text{john}, \text{pat}) \}$

$\mathcal{P} = \{ (7) \text{ischildof}(\text{john}, \text{alex}); \quad (8) \text{boy}(\text{john}); \\
(9) \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[(\text{Male} \cup \text{boy}; \text{Male})(\text{pat})], \quad \text{DL}[(\text{Male} \cup \text{hasParent})(\text{john}, \text{pat})] \}$

- **Interpretation:** $I = \{ \text{ischildof}(\text{john}, \text{alex}), \text{boy}(\text{john}), \text{hasfather}(\text{john}, \text{pat}) \}$

- **Satisfaction relation:** $I \models^\mathcal{O} \text{boy}(\text{john}); I \models^\mathcal{O} \text{DL}[; \text{hasParent}](\text{john}, \text{pat})$

  $I \models^\mathcal{O} \text{DL}[(\text{Male} \cup \text{boy}; \text{Male})(\text{pat})]$

- **Semantics:** in terms of answer sets, i.e. founded models (weak, flp, . . . )

- $I$ is a weak and flp answer set
Example: Inconsistent DL-program

\[ \Pi = \langle \mathcal{O}, \mathcal{P} \rangle \]

\[ \mathcal{O} = \left\{ \begin{array}{ll}
(1) & \text{Child} \sqsubseteq \exists \text{hasParent} \\
(2) & \text{Adopted} \sqsubseteq \text{Child} \\
(3) & \text{Female} \sqsubseteq \neg \text{Male} \\
\end{array} \right. \]

\[ \mathcal{P} = \left\{ \begin{array}{ll}
(4) & \text{Male}(\text{pat}) \\
(5) & \text{Male}(\text{john}) \\
(6) & \text{hasParent}(\text{john}, \text{pat}) \\
(7) & \text{ischildof}(\text{john}, \text{alex}); \quad \text{boy}(\text{john}); \\
(8) & \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[\text{Male} \cup \text{boy}; \text{Male]}(\text{pat}), \\
& \phantom{\leftarrow} \text{DL}[; \text{hasParent}](\text{john}, \text{pat}); \\
(9) & \perp \leftarrow \neg \text{DL}[; \text{Adopted}](\text{john}), \text{pat} \neq \text{alex}, \\
& \phantom{\leftarrow} \text{hasfather}(\text{john}, \text{pat}), \text{ischildof}(\text{john}, \text{alex}), \\
& \phantom{\leftarrow} \neg \text{DL}[\text{Child} \cup \text{boy}; \neg \text{Male}](\text{alex}) \\
\end{array} \right. \]
Example: Inconsistent DL-program

\[ \Pi = \langle \mathcal{O}, \mathcal{P} \rangle \]

\[ \mathcal{O} = \left\{ \begin{array}{l}
(1) \text{Child} \sqsubseteq \exists \text{hasParent} \\
(2) \text{Adopted} \sqsubseteq \text{Child} \\
(3) \text{Female} \sqsubseteq \neg \text{Male} \\
(7) \text{ischildof}(\text{john}, \text{alex}); \\
(8) \text{boy}(\text{john}); \\
(9) \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[\text{Male} \cup \text{boy}; \text{Male}](\text{pat}), \\
\text{DL}[; \text{hasParent}](\text{john}, \text{pat}); \\
(10) \perp \leftarrow \text{not DL}[; \text{Adopted}](\text{john}), \text{pat} \neq \text{alex}, \\
\text{hasfather}(\text{john}, \text{pat}), \text{ischildof}(\text{john}, \text{alex}), \\
\text{not DL}[	ext{Child} \cup \text{boy}; \neg \text{Male}](\text{alex})
\end{array} \right\} \]
Example: Inconsistent DL-program

$$\Pi = \langle \mathcal{O}, \mathcal{P} \rangle$$

$$\mathcal{O} = \left\{ \begin{array}{ll}
(1) & \text{Child} \sqsubseteq \exists \text{hasParent} \\
(2) & \text{Adopted} \sqsubseteq \text{Child} \\
(3) & \text{Female} \sqsubseteq \neg \text{Male} \\
\end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{ll}
(7) & \text{ischildof(john, alex)}; \\
(9) & \text{hasfather(john, pat)} \leftarrow \text{DL[Male } \sqcup \text{ boy;} \text{ Male}(pat), \\
& \text{DL[; hasParent](john, pat);} \\
(10) & \bot \leftarrow \text{not DL[; Adopted](john), pat } \neq \text{ alex,} \\
& \text{hasfather(john, pat), ischildof(john, alex),} \\
& \text{not DL[Child } \sqcup \text{ boy;} \neg \text{Male}(alex)}
\end{array} \right\}$$
Example: Inconsistent DL-program

\( \Pi = \langle \mathcal{O}, \mathcal{P} \rangle \) is inconsistent!

\[ \mathcal{O} = \left\{ \begin{array}{ll}
(1) & \text{Child } \sqsubseteq \exists \text{hasParent} \\
(2) & \text{Adopted } \sqsubseteq \text{Child} \\
(3) & \text{Female } \sqsubseteq \neg \text{Male} \\
\end{array} \right\} \]

\[ \mathcal{P} = \left\{ \begin{array}{ll}
(4) & \text{Male}(\text{pat}) \\
(5) & \text{Male}(\text{john}) \\
(6) & \text{hasParent}(\text{john}, \text{pat}) \\
(7) & \text{ischildof}(\text{john}, \text{alex}); \\
(8) & \text{boy}(\text{john}); \\
(9) & \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[\text{Male} \;\mathbb{U} \; \text{boy}; \text{Male}](\text{pat}), \\
& \quad \text{DL}; \text{hasParent}(\text{john}, \text{pat}); \\
(10) & \bot \leftarrow \text{not DL}[; \text{ Adopted}](\text{john}), \text{pat} \neq \text{alex}, \\
& \quad \text{hasfather}(\text{john}, \text{pat}), \text{ischildof}(\text{john}, \text{alex}), \\
& \quad \text{not DL}[\text{Child} \;\mathbb{U} \; \text{boy}; \neg \text{Male}](\text{alex}).
\end{array} \right\} \]

No answer sets
Example: Inconsistent DL-program

\[ \Pi = \langle O, P \rangle \text{ is consistent!} \]

\[ O = \begin{cases} 
(1) \text{Child} \sqsubseteq \exists \text{hasParent} \\
(2) \text{Adopted} \sqsubseteq \text{Child} \\
(3) \text{Female} \sqsubseteq \neg \text{Male} \\
(5) \text{Male}(\text{john}) \\
(6) \text{hasParent}(\text{john}, \text{pat}) \\
(7) \text{ischildof}(\text{john}, \text{alex}); \\
(8) \text{boy}(\text{john}); \\
(9) \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[\text{Male} \cup \text{boy}; \text{Male}](\text{pat}), \\
\text{DL}[: \text{hasParent}](\text{john}, \text{pat}); \\
(10) \bot \leftarrow \text{not DL}[: \text{Adopted}](\text{john}), \text{pat} \neq \text{alex}, \\
\text{hasfather}(\text{john}, \text{pat}), \text{ischildof}(\text{john}, \text{alex}), \\
\text{not DL}[\text{Child} \cup \text{boy}; \neg \text{Male}](\text{alex}) 
\end{cases} \]

\[ P = \begin{cases} 
(7) \text{ischildof}(\text{john}, \text{alex}); \\
(8) \text{boy}(\text{john}); \\
(9) \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[\text{Male} \cup \text{boy}; \text{Male}](\text{pat}), \\
\text{DL}[: \text{hasParent}](\text{john}, \text{pat}); \\
(10) \bot \leftarrow \text{not DL}[: \text{Adopted}](\text{john}), \text{pat} \neq \text{alex}, \\
\text{hasfather}(\text{john}, \text{pat}), \text{ischildof}(\text{john}, \text{alex}), \\
\text{not DL}[\text{Child} \cup \text{boy}; \neg \text{Male}](\text{alex}) 
\end{cases} \]

\[ I_1 = \{ \text{ischildof}(\text{john}, \text{alex}), \text{boy}(\text{john}) \} \]
Example: Inconsistent DL-program

\[ \Pi = \langle \mathcal{O}, \mathcal{P} \rangle \text{ is consistent!} \]

\[ \mathcal{O} = \{ \text{(1) } \text{Child} \sqsubseteq \exists \text{hasParent} \quad \text{(4) } \text{Male}(\text{pat}) \]
\[ \text{(2) } \text{Adopted} \sqsubseteq \text{Child} \quad \text{(5) } \text{Male}(\text{john}) \]
\[ \text{(3) } \text{Female} \sqsubseteq \neg \text{Male} \}

\[ \mathcal{P} = \{ \text{(7) } \text{ischildof}(\text{john}, \text{alex}); \quad \text{(8) } \text{boy}(\text{john}); \]
\[ \text{(9) } \text{hasfather}(\text{john}, \text{pat}) \leftarrow \text{DL}[\text{Male} \sqcup \text{boy}; \text{Male}](\text{pat}), \]
\[ \quad \text{DL}[; \text{hasParent}](\text{john}, \text{pat}); \]
\[ \text{(10) } \bot \leftarrow \text{not DL}[; \text{Adopted}](\text{john}), \text{pat} \neq \text{alex}, \]
\[ \quad \text{hasfather}(\text{john}, \text{pat}), \text{ischildof}(\text{john}, \text{alex}), \]
\[ \quad \text{not DL}[\text{Child} \sqcup \text{boy}; \neg \text{Male}](\text{alex}) \}

\[ I_1 = \{ \text{ischildof}(\text{john}, \text{alex}), \text{boy}(\text{john}) \} \]
**Ground Support Sets**

\[ d = DL[Male \cup boy; Male](pat); T = \{Female \sqsubseteq \neg Male\} \]

When is \( d \) true under interpretation \( I \)?
Ground Support Sets

\[ d = DL[\text{Male} \uplus \text{boy}; \text{Male}](\text{pat}); \quad T = \{ \text{Female} \sqsubseteq \neg \text{Male} \} \]

When is \( d \) true under interpretation \( I \)?

- \( \text{Male}(\text{pat}) \in A \)
- \( \text{boy}(\text{pat}) \in I \)
- \( \text{boy}(\text{alex}) \in I; \quad \text{Female}(\text{alex}) \in A \)
Ground Support Sets

\[ d = \text{DL}[\text{Male} \cup \text{boy}; \text{Male}](\text{pat}); \quad T_d = \{\text{Female} \sqsubseteq \neg \text{Male}; \text{Male}_{\text{boy}} \sqsubseteq \text{Male}\} \]

When is \( d \) true under interpretation \( I \)?

- \( \text{Male}(\text{pat}) \in A \)
- \( \text{Male}_{\text{boy}}(\text{pat}) \in A_d \), s.t. \( \text{boy}(\text{pat}) \in I \)
- \( \text{Male}_{\text{boy}}(\text{alex}) \in A_d \), s.t. \( \text{boy}(\text{alex}) \in I; \text{Female}(\text{alex}) \in A \)

where \( A_d = \{P_p(t) \mid P \cup p \in \lambda\} \cup \{\neg P_p(t) \mid P \cup p \in \lambda\} \)
**Ground Support Sets**

**Definition**

$S \subseteq A \cup A_d$ is a support set for $d = DL[\lambda; Q](t)$ w.r.t. $O = \langle T, A \rangle$ if either

1. $S = \{P(c)\}$ and $T_d \cup S \models Q(t)$ or
2. $S = \{P(c), P'(d)\}$, s.t. $T_d \cup S$ is inconsistent.

$\text{Supp}_O(d)$ is a set of all support sets for $d$.

\[d = DL[\text{Male} \uplus \text{boy}; \text{Male}](\text{pat}); T_d = \{\text{Female} \sqsubseteq \neg \text{Male}; \text{Male}_{\text{boy}} \sqsubseteq \text{Male}\}\]

Support sets:

- $S_1 = \{\text{Male}(\text{pat})\}$, coherent with any $I$
- $S_2 = \{\text{Male}_{\text{boy}}(\text{pat})\}$, coherent with $I \supseteq \text{boy}(\text{pat})$
- $S_3 = \{\text{Male}_{\text{boy}}(\text{alex}); \text{Female}(\text{alex})\}$, coherent with $I \supseteq \text{boy}(\text{alex})$
Ground Support Sets

Definition

\( S \subseteq A \cup A_d \) is a support set for \( d = DL[\lambda; Q](t) \) with respect to \( O = \langle T, A \rangle \) if either

(i) \( S = \{ P(c) \} \) and \( T_d \cup S \models Q(t) \) or

(ii) \( S = \{ P(c), P'(d) \} \), s.t. \( T_d \cup S \) is inconsistent.

\( \text{Supp}_O(d) \) is a set of all support sets for \( d \).

\[ l \models^O d \text{ iff there exists } S \in \text{Supp}_O(d), \text{ which is coherent with } l. \]
Nonground Support Sets

\[ d = DL[Male \uplus boy; Male](pat), \quad T_d = \{ Female \sqsubseteq \neg Male; Male_{boy} \sqsubseteq Male \} \]

Support sets:

- \( S_1 = \{ Male(pat) \} \)
- \( S_2 = \{ Male_{boy}(pat) \} \)
- \( S_3 = \{ Male_{boy}(c); Female(c) \} \quad c \in C \)
Nonground Support Sets

\[ d = DL[Male \cup boy; Male](X), \quad T_d = \{ Female \sqsubseteq \neg Male; Male_{boy} \sqsubseteq Male \} \]

Nonground support sets:

- \( S_1 = \{ Male(X) \} \)
- \( S_2 = \{ Male_{boy}(X) \} \)
- \( S_3 = \{ Male_{boy}(Y); Female(Y) \} \)
Nonground Support Sets

Definition

\[ S = \{ P(Y), P'(Y') \} \] (\( S = \{ P(Y) \} \)) is a nonground support set for a DL-atom \( d(X) \) w.r.t. \( T \) if for every \( \theta : V \rightarrow C \) it holds that \( S\theta \) is a support set for \( d(X\theta) \) w.r.t. \( O_C = \langle T, A_C \rangle \), where \( A_C \) is a set of all possible assertions over \( C \).

\[
d = DL[Male \uplus boy; Male](X), \ T_d = \{ Female \sqsubseteq \neg Male; Male_{boy} \sqsubseteq Male \}
\]

Nonground support sets:

- \( S_1 = \{ Male(X) \} \)
- \( S_2 = \{ Male_{boy}(X) \} \)
- \( S_3 = \{ Male_{boy}(Y); Female(Y) \} \)
Nonground Support Sets

Definition

\[ S = \{ P(Y), P'(Y') \} \quad (S = \{ P(Y) \}) \] is a nonground support set for a DL-atom \( d(X) \) w.r.t. \( T \) if for every \( \theta : V \rightarrow C \) it holds that \( S\theta \) is a support set for \( d(X\theta) \) w.r.t. \( O_C = \langle T, A_C \rangle \), where \( A_C \) is a set of all possible assertions over \( C \).

Nonground support sets are compact representations of ground ones.
Nonground Support Sets

Definition

$S = \{P(Y), P'(Y')\}$ ($S = \{P(Y)\}$) is a nonground support set for a DL-atom $d(X)$ w.r.t. $T$ if for every $\theta : V \rightarrow C$ it holds that $S\theta$ is a support set for $d(X\theta)$ w.r.t. $O_C = \langle T, A_C \rangle$, where $A_C$ is a set of all possible assertions over $C$.

Nonground support sets are **compact representations** of ground ones.

Completeness: family of nonground support sets $S$ for $d(X)$ is complete w.r.t. $O$ if for every $\theta : X \rightarrow C$ and $S \in \text{Supp}_O(d(X\theta))$ some $S' \in S$ exists, s.t. $S = S'\theta'$.

Complete support families allow to **avoid access** to $O$ during DL-atom evaluation.
Nonround Support Set Computation

\( d = DL[Male \uplus boy; Male](X); T = \{ Female \sqsubseteq \neg Male \} \)

- **Construct** \( T_d \):

- **Compute classification** \( Cl(T_d) \) (e.g. using ASP techniques):

- **Extract support sets from** \( Cl(T_d) \):
Nonround Support Set Computation

\[ d = \text{DL}[\text{Male} \cup \text{boy}; \text{Male}](X); \ T = \{\text{Female} \subseteq \neg \text{Male}\} \]

- **Construct** \( \mathcal{T}_d \):
  \[ \mathcal{T}_d = \mathcal{T} \cup \{\text{Male}_\text{boy} \subseteq \text{Male}\} \]

- **Compute classification** \( \text{Cl}(\mathcal{T}_d) \) (e.g. using ASP techniques):

- **Extract support sets from** \( \text{Cl}(\mathcal{T}_d) \):

  \[ \begin{align*}
  S_1 &= \{\text{Male}(X)\} \\
  S_2 &= \{\text{Male}_\text{boy}(X)\} \\
  S_3 &= \{\text{Male}_\text{boy}(Y), \neg \text{Male}(Y)\} \\
  S_4 &= \{\text{Male}_\text{boy}(Y), \text{Female}(Y)\} \\
  S_5 &= \{\text{Male}(Y), \neg \text{Male}(Y)\} \\
  S_6 &= \{\text{Male}(Y), \text{Female}(Y)\}
  \end{align*} \]
Nonround Support Set Computation

\( d = DL[\text{Male} \sqcup \text{boy}; \text{Male}] (X); T = \{ \text{Female} \sqsubseteq \neg \text{Male} \} \)

- Construct \( T_d \):
  \[
  T_d = T \cup \{ \text{Male}_{\text{boy}} \sqsubseteq \text{Male} \}
  \]

- Compute classification \( Cl(T_d) \) (e.g. using ASP techniques):
  \[
  cl(T_d) = T_d \cup \{ \text{Male} \sqsubseteq \neg \text{Female}; \text{Male}_{\text{boy}} \sqsubseteq \neg \text{Female} \} \cup \{ P \sqsubseteq P \mid P \in P \}
  \]

- Extract support sets from \( Cl(T_d) \):
  \[
  S_1 = \{ \text{Male} (X) \}
  S_2 = \{ \text{Male}_{\text{boy}} (X) \}
  S_3 = \{ \text{Male}_{\text{boy}} (Y), \neg \text{Male} (Y) \}
  S_4 = \{ \text{Male}_{\text{boy}} (Y), \text{Female} (Y) \}
  \]
Nonround Support Set Computation

\[ d = DL[Male \cup boy; \text{Male}](X); T = \{Female \sqsubseteq \neg Male\} \]

- **Construct** \( T_d \):
  \[ T_d = T \cup \{\text{Male}_{\text{boy}} \sqsubseteq \text{Male}\} \]

- **Compute classification** \( Cl(T_d) \) (e.g. using ASP techniques):
  \[ cl(T_d) = T_d \cup \{\text{Male} \sqsubseteq \neg Female; \text{Male}_{\text{boy}} \sqsubseteq \neg Female\} \cup \{P \sqsubseteq P \mid P \in P\} \]

- **Extract support sets from** \( Cl(T_d) \):
  - \( S_1 = \{\text{Male}(X)\} \)
  - \( S_2 = \{\text{Male}_{\text{boy}}(X)\} \)
  - \( S_3 = \{\text{Male}_{\text{boy}}(Y), \neg \text{Male}(Y)\} \)
  - \( S_4 = \{\text{Male}_{\text{boy}}(Y), \text{Female}(Y)\} \)
  - \( S_5 = \{\text{Male}(Y), \neg \text{Male}(Y)\} \)
  - \( S_6 = \{\text{Male}(Y), \text{Female}(Y)\} \)
Nonround Support Set Computation

d = DL[Male ∪ boy; Male](X); \mathcal{T} = \{Female \sqsubseteq \neg Male\}

- Construct \( \mathcal{T}_d \):
  \[ \mathcal{T}_d = \mathcal{T} \cup \{Male_{\text{boy}} \sqsubseteq \text{Male}\} \]

- Compute classification \( Cl(\mathcal{T}_d) \) (e.g. using ASP techniques):
  \[ cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{\text{Male} \sqsubseteq \neg \text{Female}; \ Male_{\text{boy}} \sqsubseteq \neg \text{Female}\} \cup \{P \sqsubseteq P \mid P \in P\} \]

- Extract suport sets from \( Cl(\mathcal{T}_d) \):
  \[ S_1 = \{Male(X)\} \]
  \[ S_2 = \{Male_{\text{boy}}(X)\} \]
  \[ S_3 = \{Male_{\text{boy}}(Y), \neg \text{Male}(Y)\} \]
  \[ S_4 = \{Male_{\text{boy}}(Y), \text{Female}(Y)\} \]
  \[ S_5 = \{\text{Male}(Y), \neg \text{Male}(Y)\} \]
  \[ S_6 = \{\text{Male}(Y), \text{Female}(Y)\} \]
Nonround Support Set Computation

\(d = DL[Male \cup boy; Male](X); \mathcal{T} = \{Female \sqsubseteq \neg Male\}\)

- Construct \(\mathcal{T}_d\):
  \[\mathcal{T}_d = \mathcal{T} \cup \{Male_{boy} \sqsubseteq Male\}\]

- Compute classification \(Cl(\mathcal{T}_d)\) (e.g. using ASP techniques):
  \[cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{Male \sqsubseteq \neg Female; Male_{boy} \sqsubseteq \neg Female\} \cup \{P \sqsubseteq P \mid P \in P\}\]

- Extract support sets from \(Cl(\mathcal{T}_d)\):
  - \(S_1 = \{Male(X)\}\)
  - \(S_2 = \{Male_{boy}(X)\}\)
  - \(S_3 = \{Male_{boy}(Y), \neg Male(Y)\}\)
  - \(S_4 = \{Male_{boy}(Y), Female(Y)\}\)
  - \(S_5 = \{Male(Y), \neg Male(Y)\}\)
  - \(S_6 = \{Male(Y), Female(Y)\}\) \(\mathcal{O}\) is consistent!
Nonround Support Set Computation

\[
d = DL[\text{Male} \sqcup \text{boy}; \text{Male}](X); \mathcal{T} = \{\text{Female} \sqsubseteq \neg\text{Male}\}
\]

- Construct \( \mathcal{T}_d \):
  \[
  \mathcal{T}_d = \mathcal{T} \cup \{\text{Male}_{\text{boy}} \sqsubseteq \text{Male}\}
  \]

- Compute classification \( Cl(\mathcal{T}_d) \) (e.g. using ASP techniques):
  \[
  cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{\text{Male} \sqsubseteq \neg\text{Female}; \text{Male}_{\text{boy}} \sqsubseteq \neg\text{Female}\} \cup \{P \sqsubseteq P \mid P \in \mathcal{P}\}
  \]

- Extract support sets from \( Cl(\mathcal{T}_d) \):
  - \( S_1 = \{\text{Male}(X)\} \)
  - \( S_2 = \{\text{Male}_{\text{boy}}(X)\} \)
  - \( S_3 = \{\text{Male}_{\text{boy}}(Y), \neg\text{Male}(Y)\} \)
  - \( S_4 = \{\text{Male}_{\text{boy}}(Y), \text{Female}(Y)\} \)

\( \{S_1, S_2, S_3, S_4\} \) is complete!
Repair Answer Set Computation

✓ Compute complete support families $S$ for all DL-atoms of $\Pi$
  
  • Construct $\hat{\Pi}$ from $\Pi = \langle O, P \rangle$:
    1. Replace all DL-atoms $a$ with normal atoms $e_a$
    2. Add guessing rules on values of $a$: $e_a \vee ne_a$
  
  • For all $\hat{I} \in AS(\hat{\Pi})$:
    $D_p = \{ a \mid e_a \in \hat{I} \}$; $D_n = \{ a \mid ne_a \in \hat{I} \}$

✓ Ground support sets in $S$ wrt. $\hat{I}$ and $A$: $S^\hat{I}_{gr} \leftarrow Gr(S, \hat{I}, A)$

✓ Find $A'$, such that
  
  ✓ For all $a \in D_p$: there is $S \in S^\hat{I}_{gr}(a)$, s.t.
  $S \cap A' \neq \emptyset$ or $S \subseteq A_a$
  
  ✓ For all $a' \in D_n$: for all $S \in S^\hat{I}_{gr}(a')$:
  $S \cap A' = \emptyset$ and $S \not\subseteq A_{a'}$

✓ Minimality check of $\hat{I}|_{\Pi}$ wrt. $\Pi' = \langle O', P \rangle$, $O' = \langle T, A' \rangle$
Algorithm 1: $SupRAnsSet$: all deletion repair answer sets

**Input:** $\Pi = \langle T \cup A, P \rangle$

**Output:** $flpRAS(\Pi)$

(a) compute a complete set $S$ of nongr. supp. sets for the DL-atoms in $\Pi$

(b) for $\hat{I} \in AS(\hat{\Pi})$

(c) if $S_{gr}^{\hat{I}}(a) \neq \emptyset$ for $a \in D_p$ and every $S \in S_{gr}^{\hat{I}}(a)$ for $a \in D_n$ fulfills $S \cap A \neq \emptyset$ then

(d) for all $a \in D_p$

(e) if some $S \in S_{gr}^{\hat{I}}(a)$ exists s.t. $S \cap A = \emptyset$ then pick next $a$

(f) else remove each $S$ from $S_{gr}^{\hat{I}}(a)$ s.t. $S \cap A \cap \bigcup_{a' \in D_n} S_{gr}^{\hat{I}}(a') \neq \emptyset$

(g) $A' \leftarrow A \setminus \bigcup_{a' \in D_n} S_{gr}^{\hat{I}}(a')$

(h) if $flpFND(\hat{I}, \langle T \cup A', P \rangle)$ then output $\hat{I}|_{\Pi}$

end
Algorithm 1: SupRAnsSet: all deletion repair answer sets

Input: $\Pi = \langle \mathcal{T} \cup \mathcal{A}, \mathcal{P} \rangle$
Output: $\text{flipRAS}(\Pi)$

(a) compute a complete set $\mathcal{S}$ of nongr. supp. sets for the DL-atoms in $\Pi$
(b) for $\hat{I} \in \text{AS}^*(\Pi)$ do

SupRAnsSet is sound and complete wrt. deletion repair answer sets!

if some $S \in S_{gr}^\hat{i}(a)$ exists s.t. $S \cap \mathcal{A} = \emptyset$ then pick next $a$
else remove each $S$ from $S_{gr}^\hat{i}(a)$ s.t. $S \cap \mathcal{A} \cap \bigcup_{a' \in D_n} S_{gr}^\hat{i}(a') \neq \emptyset$

(f) if $S_{gr}^\hat{i}(a) = \emptyset$ then pick next $\hat{I}$

end

(g) $\mathcal{A}' \leftarrow \mathcal{A} \setminus \bigcup_{a' \in D_n} S_{gr}^\hat{i}(a')$

if $\text{flipFND}(\hat{I}, \langle \mathcal{T} \cup \mathcal{A}', \mathcal{P} \rangle)$ then output $\hat{I}\|\Pi$

end
Experiments

![Graph 1](image1)

![Graph 2](image2)
Related Work

**Inconsistencies in $DL$-Lite$_{\mathcal{A}}$ ontologies:**

- Consistent query answering over $DL$-Lite ontologies based on repair technique [Lembo et al., 2010], [Bienvenu, 2012]

- QA to $DL$-Lite$_{\mathcal{A}}$ ontologies that miss expected tuples (abductive explanations corresponding to repairs) [Calvanese et al., 2012]

**Support sets in other works:**

- Support sets for $\text{HEX}$-programs [Eiter et al, AAAI’2014] as more abstract structures
Conclusion and Future Work

Conclusions:

- Ground and nonground support sets for DL-atoms
  - Allow evaluation of DL-atoms avoiding ontology access
- Support sets for $DL$-$Lite_A$ are small and efficiently computable
- Effective sound and complete algorithm $SupRA$nsSet for deletion repair computation based on support sets
- Implementation in DLVHEX and evaluation on a set of benchmarks

Further and future work:

- Extensions to other DLs (e.g. $EL$)
- Computing preferred repairs
  (e.g. $\sigma$-selection [Eiter et al, IJCAI’2013])

Diego Calvanese, Magdalena Ortiz, Mantas Simkus, and Giorgio Stefanoni.

Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo.
DL-program: syntax

Signature: $\Sigma = \langle C, I, P, C, R \rangle$, where
- $\Sigma_0 = \langle I, C, R \rangle$ is a DL signature;
- $C \supseteq I$ is a set of constant symbols;
- $P$ is a finite set of predicate symbols of arity $\geq 0$, s.t. $P \cap \{ C \cup R \} = \emptyset$.

DL-atom is of the form $DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](t)$, $m \geq 0$, where
- $S_i \in C \cup R$;
- $op_i \in \{ \lor, \lor, \land \}$;
- $p_i \in P$ (unary or binary);
- $Q(t)$ is a DL-query:
  - $C(t_1), \neg C(t_1), t = t_1$, where $C \in C$;
  - $R(t_1, t_2), \neg R(t_1, t_2), t = t_1, t_2$, where $R \in R$.
  - $C \sqsubseteq D, C \not\sqsubseteq D, t = \epsilon$, where $C, D \in C \cup \{ \top, \bot \}$;

DL-program: $\Pi = \langle O, P \rangle$, $O$ is a DL ontology, $P$ is a set of DL-rules:

\[ a_1 \lor \ldots \lor a_n \leftarrow b_1, \ldots b_k, \text{not } b_{k+1}, \ldots, \text{not } b_m, \]

$m \geq k \geq 0$, $a_i$ is a classical literal; $b_i$ is a classical literal or a DL-atom.
Consider grounding $\text{grd}(\Pi) = \langle \mathcal{O}, \text{grd}(P) \rangle$ of $\Pi = \langle \mathcal{O}, P \rangle$ over $\mathcal{C}$ and $\mathcal{P}$.

Interpretation $I$ is a consistent set of ground literals over $\mathcal{C}$ and $\mathcal{P}$.

- for ground literal $\ell$: $I \models O \ell$ iff $\ell \in I$;
- for ground DL-atom $a = DL[S_1 \text{op}_1 p_1, \ldots, S_m \text{op}_m p_m; Q](c)$:
  $$I \models O a$$
  iff $\tau(\langle T, A \cup \lambda^l(a) \rangle) \models Q(c)$, where $\tau(O)$ is a modular translation of $O$ to FOL, $\lambda^l(a) = \bigcup_{i=1}^{m} A_i(l)$ is a DL-update of $O$ under $I$ by $a$:
  - $A_i(I) = \{ S_i(t) \mid p_i(t) \in I \}$, for $\text{op}_i = \cup$;
  - $A_i(I) = \{ \neg S_i(t) \mid p_i(t) \in I \}$, for $\text{op}_i = \cup$;
  - $A_i(I) = \{ \neg S_i(t) \mid p_i(t) \notin I \}$, for $\cap$.

FLP-reduct $\rho_{\text{flp}}P^I$ of $P$ is a set of ground DL-rules $r$ s.t. $I \models b^+(r)$, $I \not\models b^-(r)$.

Weak-reduct $\rho_{\text{weak}}P^I$ of $P$: removes all DL-atoms $b_i$, $1 \leq i \leq k$ and all not $b_j$, $k < j \leq m$ from the rules of $\rho_{\text{flp}}P^I$.

$I$ is an $x$-answer set of $P$ iff $I$ is a minimal model of its $x$-reduct.
Network Benchmark

$$\mathcal{O} = \left\{ \begin{array}{l}
(1) \: \exists \text{forbid} \sqsubseteq \text{Block} \\
(2) \: \text{Broken} \sqsubseteq \text{Block} \\
(3) \: \text{Block} \sqsubseteq \neg \text{Avail} \\
(4) \: \text{edge}(n_i, n_j) \\
(5) \: \ldots \\
(6) \: \ldots 
\end{array} \right\}$$

$$\mathcal{P}_{\text{guess}} = \left\{ \begin{array}{l}
(1) \: \text{go}(X, Y) \leftarrow \text{open}(X), \text{open}(Y), \text{DL}[; \text{edge}](X, Y). \\
(2) \: \text{route}(X, Z) \leftarrow \text{route}(X, Y), \text{route}(Y, Z). \\
(3) \: \text{route}(X, Y) \leftarrow \neg \text{DL}[\text{Block} \uplus \text{block}; \text{forbid}](X, Y), \text{go}(X, Y). \\
(4) \: \text{open}(X) \lor \text{block}(X) \leftarrow \neg \text{DL}[; \neg \text{Avail}](X), \text{node}(X). \\
(5) \: \text{negls}(X) \leftarrow \text{node}(X), \text{route}(X, Y), X \neq Y. \\
(6) \: \bot \leftarrow \text{node}(X), \neg \text{negls}(X). 
\end{array} \right\}$$
Network Benchmark

\[ \mathcal{O} = \left\{ \begin{array}{l}
(1) \exists \text{forbid} \sqsubseteq \text{Block} \\
(2) \text{Broken} \sqsubseteq \text{Block} \\
(3) \text{Block} \sqsubseteq \neg \text{Avail} \\
(4) \text{edge}(n_i, n_j)
\end{array} \right\} \]

\[ \mathcal{P}_{\text{con}} = \left\{ \begin{array}{l}
(1) \text{go}(X, Y) \leftarrow \text{open}(X), \text{open}(Y), \text{DL}[; \text{edge}](X, Y).
\end{array} \right\} \]

\[ \left\{ \begin{array}{l}
(2) \text{route}(X, Z) \leftarrow \text{route}(X, Y), \text{route}(Y, Z).
\end{array} \right\} \]

\[ \left\{ \begin{array}{l}
(3') \text{route}(X, Y) \leftarrow \text{go}(X, Y), \neg \text{DL}[; \text{forbid}](X, Y).
\end{array} \right\} \]

\[ \left\{ \begin{array}{l}
(4') \text{open}(X) \leftarrow \text{node}(X), \neg \text{DL}[; \neg \text{Avail}](X).
\end{array} \right\} \]

\[ \left\{ \begin{array}{l}
(5) \text{negls}(X) \leftarrow \text{node}(X), \text{route}(X, Y), X \neq Y.
\end{array} \right\} \]

\[ \left\{ \begin{array}{l}
(6') \perp \leftarrow \text{in}(X), \text{out}(Y), \neg \text{route}(X, Y).
\end{array} \right\} \]