Towards Practical Deletion Repair of Inconsistent DL-programs

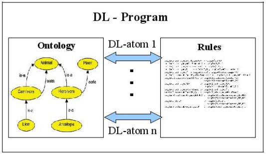
Thomas Eiter Michael Fink Daria Stepanova

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

ECAI 2014 –August, 21, 2014

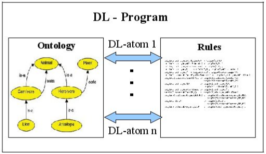
Motivation

- 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

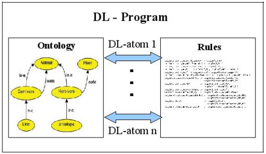


However, information exchange between $\mathcal{P}$ and $\mathcal{O}$ can cause **inconsistency** of the DL-program (absence of answer sets).

! [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 $\mathcal{O}$ + 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**.

In this work: Algorithm for DL-program repair based on support sets for DL-atoms. Effective for ontologies in *DL-Lite*$_{\mathcal{A}}$.

Overview

Motivation

DL-programs

Support Sets for DL-atoms

Repair Answer Set Computation

Experiments

Conclusion

DL-Lite$_{\mathcal{A}}$

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

$$C \rightarrow A \mid \exists R \qquad R \rightarrow P|P^-$$

- 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

$$C_1 \sqsubseteq C_2, \qquad C_1 \sqsubseteq \neg C_2,$$
$$R_1 \sqsubseteq R_2, \qquad R_1 \sqsubseteq \neg R_2, \qquad (funct\ R)$$

  - ABox $\mathcal{A}$ specifying the facts that hold in the domain

$$A(b) \qquad P(a, b)$$

$\mathit{DL\text{-}Lite}_{\mathcal{A}}$

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

$$C \rightarrow A \mid \exists R \qquad R \rightarrow P \mid P^{-}$$

- A $\mathit{DL\text{-}Lite}_{\mathcal{A}}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ consists of:
  - TBox $\mathcal{T}$ specifying constraints at the conceptual level

$$\begin{array}{lll} C_1 \sqsubseteq C_2, & C_1 \sqsubseteq \neg C_2, & \\ R_1 \sqsubseteq R_2, & R_1 \sqsubseteq \neg R_2, & (\mathit{funct}\ R) \end{array}$$

  - ABox $\mathcal{A}$ specifying the facts that hold in the domain

$$A(b) \qquad P(a, b)$$

Example

$$\mathcal{T} = \left\{ \begin{array}{l} \mathit{Child} \sqsubseteq \exists \mathit{hasParent} \\ \mathit{Female} \sqsubseteq \neg \mathit{Male} \end{array} \right\} \qquad \mathcal{A} = \left\{ \begin{array}{l} \mathit{hasParent}(\mathit{john}, \mathit{pat}) \\ \mathit{Male}(\mathit{john}) \end{array} \right\}$$

DL-Lite$_{\mathcal{A}}$

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

$$C \rightarrow A \mid \exists R \qquad R \rightarrow P|P^-$$

- 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

$$C_1 \sqsubseteq C_2, \qquad C_1 \sqsubseteq \neg C_2,$$
$$R_1 \sqsubseteq R_2, \qquad R_1 \sqsubseteq \neg R_2, \qquad (funct\ R)$$

  - ABox $\mathcal{A}$ specifying the facts that hold in the domain

$$A(b) \qquad P(a, b)$$

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

[Calvanese *et al.*, 2007]

Example: DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists \mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

Example: DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists \mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ \mathit{ischildof}(\mathit{john}, \mathit{alex});\quad (8)\ \mathit{boy}(\mathit{john}); \\ (9)\ \mathit{hasfather}(\mathit{john}, \mathit{pat}) \leftarrow \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}), \\ \qquad\qquad\qquad\qquad\qquad \mathrm{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat}) \end{array} \right\}$$

Example: DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists \mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ \mathit{ischildof}(\mathit{john}, \mathit{alex});\quad (8)\ \mathit{boy}(\mathit{john}); \\ (9)\ \mathit{hasfather}(\mathit{john}, \mathit{pat}) \leftarrow \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}), \\ \qquad\qquad\qquad\qquad\qquad\quad \mathrm{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat}) \end{array} \right\}$$

- Interpretation: $I = \{\mathit{ischildof}(\mathit{john}, \mathit{alex}), \mathit{boy}(\mathit{john}), \mathit{hasfather}(\mathit{john}, \mathit{pat})\}$
- Satisfaction relation: $I \models^{\mathcal{O}} \mathit{boy}(\mathit{john})$; $I \models^{\mathcal{O}} \mathit{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat})$
  $I \models^{\mathcal{O}} \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat})$
- Semantics: in terms of answer sets, i.e. founded models (weak, flp, ...)
- I is a weak and flp answer set

Example: Inconsistent DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists \mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ \mathit{ischildof}(\mathit{john}, \mathit{alex});\quad (8)\ \mathit{boy}(\mathit{john}); \\ (9)\ \mathit{hasfather}(\mathit{john}, \mathit{pat}) \leftarrow \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}), \\ \qquad\qquad\qquad\qquad\qquad \mathrm{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat}); \\ (10)\ \bot \leftarrow \mathit{not}\ \mathrm{DL}[; \mathit{Adopted}](\mathit{john}), \mathit{pat} \neq \mathit{alex}, \\ \qquad\qquad \mathit{hasfather}(\mathit{john}, \mathit{pat}), \mathit{ischildof}(\mathit{john}, \mathit{alex}), \\ \qquad\qquad \mathit{not}\ \mathrm{DL}[\mathit{Child} \uplus \mathit{boy}; \neg \mathit{Male}](\mathit{alex}) \end{array} \right\}$$

Example: Inconsistent DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists \mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ \mathit{ischildof}(\mathit{john}, \mathit{alex});\quad (8)\ \mathit{boy}(\mathit{john}); \\ (9)\ \mathit{hasfather}(\mathit{john}, \mathit{pat}) \leftarrow \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}), \\ \qquad\qquad \mathrm{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat}); \\ (10)\ \bot \leftarrow \mathit{not}\ \mathrm{DL}[; \mathit{Adopted}](\mathit{john}), \mathit{pat} \neq \mathit{alex}, \\ \qquad \mathit{hasfather}(\mathit{john}, \mathit{pat}), \mathit{ischildof}(\mathit{john}, \mathit{alex}), \\ \qquad \mathit{not}\ \mathrm{DL}[\mathit{Child} \uplus \mathit{boy}; \neg \mathit{Male}](\mathit{alex}) \end{array} \right\}$$

Example: Inconsistent DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists \mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ \mathit{ischildof}(\mathit{john}, \mathit{alex});\quad (8)\ \mathit{boy}(\mathit{john}); \\ (9)\ \mathit{hasfather}(\mathit{john}, \mathit{pat}) \leftarrow \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}), \\ \qquad \mathrm{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat}); \\ (10)\ \bot \leftarrow \mathit{not}\ \mathrm{DL}[; \mathit{Adopted}](\mathit{john}), \mathit{pat} \neq \mathit{alex}, \\ \qquad \mathit{hasfather}(\mathit{john}, \mathit{pat}), \mathit{ischildof}(\mathit{john}, \mathit{alex}), \\ \qquad \mathit{not}\ \mathrm{DL}[\mathit{Child} \uplus \mathit{boy}; \neg \mathit{Male}](\mathit{alex}) \end{array} \right\}$$

Example: Inconsistent DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \mathit{Child} \sqsubseteq \exists \mathit{hasParent} & (4)\ \mathit{Male}(\mathit{pat}) \\ (2)\ \mathit{Adopted} \sqsubseteq \mathit{Child} & (5)\ \mathit{Male}(\mathit{john}) \\ (3)\ \mathit{Female} \sqsubseteq \neg \mathit{Male} & (6)\ \mathit{hasParent}(\mathit{john}, \mathit{pat}) \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ \mathit{ischildof}(\mathit{john}, \mathit{alex});\quad (8)\ \mathit{boy}(\mathit{john}); \\ (9)\ \mathit{hasfather}(\mathit{john}, \mathit{pat}) \leftarrow \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}), \\ \qquad\qquad\qquad\qquad\qquad\quad \mathrm{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat}); \\ (10)\ \bot \leftarrow \mathit{not}\ \mathrm{DL}[; \mathit{Adopted}](\mathit{john}), \mathit{pat} \neq \mathit{alex}, \\ \qquad\qquad \mathit{hasfather}(\mathit{john}, \mathit{pat}), \mathit{ischildof}(\mathit{john}, \mathit{alex}), \\ \qquad\qquad \mathit{not}\ \mathrm{DL}[\mathit{Child} \uplus \mathit{boy}; \neg \mathit{Male}](\mathit{alex}). \end{array} \right\}$$

No answer sets

Example: Inconsistent DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ Child \sqsubseteq \exists hasParent & \\ (2)\ Adopted \sqsubseteq Child & (5)\ Male(john) \\ (3)\ Female \sqsubseteq \neg Male & (6)\ hasParent(john, pat) \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ ischildof(john, alex); \quad (8)\ boy(john); \\ (9)\ hasfather(john, pat) \leftarrow \text{DL}[Male \uplus boy; Male](pat), \\ \qquad\qquad\qquad\qquad\qquad\quad \text{DL}[; hasParent](john, pat); \\ (10)\ \bot \leftarrow not\ \text{DL}[; Adopted](john), pat \neq alex, \\ \qquad\qquad hasfather(john, pat), ischildof(john, alex), \\ \qquad\qquad not\ \text{DL}[Child \uplus boy; \neg Male](alex) \end{array} \right\}$$

$I_1 = \{ischildof(john, alex), boy(john)\}$

Example: Inconsistent DL-program

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

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ Child \sqsubseteq \exists hasParent & (4)\ Male(pat) \\ (2)\ Adopted \sqsubseteq Child & (5)\ Male(john) \\ (3)\ Female \sqsubseteq \neg Male & \end{array} \right\}$$

$$\mathcal{P} = \left\{ \begin{array}{l} (7)\ ischildof(john, alex); \quad (8)\ boy(john); \\ (9)\ hasfather(john, pat) \leftarrow \mathrm{DL}[Male \uplus boy; Male](pat), \\ \qquad\qquad\qquad\qquad\qquad\quad \mathrm{DL}[;hasParent](john, pat); \\ (10)\ \bot \leftarrow not\ \mathrm{DL}[;Adopted](john), pat \neq alex, \\ \qquad\qquad hasfather(john, pat), ischildof(john, alex), \\ \qquad\qquad not\ \mathrm{DL}[Child \uplus boy; \neg Male](alex) \end{array} \right\}$$

$I_1 = \{ischildof(john, alex), boy(john)\}$

Ground Support Sets

$d = \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}); \mathcal{T} = \{\mathit{Female} \sqsubseteq \neg \mathit{Male}\}$

When is *d* true under interpretation *I*?

Ground Support Sets

$d = \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat}); \mathcal{T} = \{\mathit{Female} \sqsubseteq \neg \mathit{Male}\}$

When is d true under interpretation I?

- $\mathit{Male}(\mathit{pat}) \in \mathcal{A}$
- $\mathit{boy}(\mathit{pat}) \in I$
- $\mathit{boy}(\mathit{alex}) \in I$; $\mathit{Female}(\mathit{alex}) \in \mathcal{A}$

Ground Support Sets

$$d = \mathrm{DL}[\overbrace{Male \uplus boy}^{\lambda}; Male](pat); \mathcal{T}_d = \{Female \sqsubseteq \neg Male; Male_{boy} \sqsubseteq Male\}$$

When is d true under interpretation I?

- $Male(pat) \in \mathcal{A}$
- $Male_{boy}(pat) \in \mathcal{A}_d$, s.t. $boy(pat) \in I$
- $Male_{boy}(alex) \in \mathcal{A}_d$, s.t. $boy(alex) \in I$; $Female(alex) \in \mathcal{A}$

where $\mathcal{A}_d = \{P_p(\mathbf{t}) \mid P \uplus p \in \lambda\} \cup \{\neg P_p(\mathbf{t}) \mid P \cup\!\!\!- p \in \lambda\}$

Ground Support Sets

Definition

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

(i) $S = \{P(\mathbf{c})\}$ and $\mathcal{T}_d \cup S \models Q(\mathbf{t})$ or

(ii) $S = \{P(\mathbf{c}), P'(\mathbf{d})\}$, s.t. $\mathcal{T}_d \cup S$ is inconsistent.

$\mathit{Supp}_{\mathcal{O}}(d)$ is a set of all support sets for d.

$d = \mathrm{DL}[\mathit{Male} \uplus \mathit{boy}; \mathit{Male}](\mathit{pat})$; $\mathcal{T}_d = \{\mathit{Female} \sqsubseteq \neg \mathit{Male}; \mathit{Male}_{\mathit{boy}} \sqsubseteq \mathit{Male}\}$

Support sets:

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

Ground Support Sets

Definition

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

(i) $S = \{P(\mathbf{c})\}$ and $\mathcal{T}_d \cup S \models Q(\mathbf{t})$ or

(ii) $S = \{P(\mathbf{c}), P'(\mathbf{d})\}$, s.t. $\mathcal{T}_d \cup S$ is inconsistent.

$Supp_{\mathcal{O}}(d)$ is a set of all support sets for d.

$I \models^{\mathcal{O}} d$ iff there exists $S \in Supp_{\mathcal{O}}(d)$, which is coherent with I.

Nonground Support Sets

$d = \mathrm{DL}[Male \uplus boy; Male](pat)$, $\mathcal{T}_d = \{Female \sqsubseteq \neg Male; Male_{boy} \sqsubseteq Male\}$

Support sets:

- $S_1 = \{Male(pat)\}$
- $S_2 = \{Male_{boy}(pat)\}$
- $S_3 = \{Male_{boy}(c); Female(c)\}$ $c \in \mathcal{C}$

Nonground Support Sets

$d = \mathrm{DL}[Male \uplus boy; Male](X)$, $\mathcal{T}_d = \{Female \sqsubseteq \neg Male; Male_{boy} \sqsubseteq Male\}$

Nonground support sets:

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

Nonground Support Sets

Definition

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

$d = \mathrm{DL}[Male \uplus boy; Male](X)$, $\mathcal{T}_d = \{Female \sqsubseteq \neg Male; Male_{boy} \sqsubseteq Male\}$

Nonground support sets:

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

Nonground Support Sets

Definition

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

Nonground support sets are compact representations of ground ones.

Nonground Support Sets

Definition

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

Nonground support sets are compact representations of ground ones.

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

Complete support families alow to avoid access to $\mathcal{O}$ during DL-atom evaluation.

Nonround Support Set Computation

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

- Construct $\mathcal{T}_d$:

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

- Extract suport sets from $Cl(\mathcal{T}_d)$:

Nonround Support Set Computation

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

- 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):

- Extract suport sets from $Cl(\mathcal{T}_d)$:

Nonround Support Set Computation

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

- 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):
  $cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{Male \sqsubseteq \neg Female;\ Male_{boy} \sqsubseteq \neg Female\} \cup \{P \sqsubseteq P \mid P \in \mathbf{P}\}$

- Extract suport sets from $Cl(\mathcal{T}_d)$:

Nonround Support Set Computation

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

- 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):
  $cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{Male \sqsubseteq \neg Female;\ Male_{boy} \sqsubseteq \neg Female\} \cup \{P \sqsubseteq P \mid P \in \mathbf{P}\}$

- Extract suport sets from $Cl(\mathcal{T}_d)$:
  - $S_1 = \{Male(X)\}$
  - $S_2 = \{Male_{boy}(X)\}$
  - $S_3 = \{Male_{boy}(Y), \neg Male(Y)\}$
  - $S_4 = \{Male_{boy}(Y), Female(Y)\}$
  - $S_5 = \{Male(Y), \neg Male(Y)\}$
  - $S_6 = \{Male(Y), Female(Y)\}$

Nonround Support Set Computation

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

- 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):
  $cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{Male \sqsubseteq \neg Female;\ Male_{boy} \sqsubseteq \neg Female\} \cup \{P \sqsubseteq P \mid P \in \mathbf{P}\}$

- Extract suport sets from $Cl(\mathcal{T}_d)$:
  - $S_1 = \{Male(X)\}$
  - $S_2 = \{Male_{boy}(X)\}$
  - $S_3 = \{Male_{boy}(Y), \neg Male(Y)\}$
  - $S_4 = \{Male_{boy}(Y), Female(Y)\}$
  - $S_5 = \{Male(Y), \neg Male(Y)\}$
  - $S_6 = \{Male(Y), Female(Y)\}$

Nonround Support Set Computation

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

- 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):
  $cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{Male \sqsubseteq \neg Female;\ Male_{boy} \sqsubseteq \neg Female\} \cup \{P \sqsubseteq P \mid P \in \mathbf{P}\}$

- Extract suport 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

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

- 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):
  $cl(\mathcal{T}_d) = \mathcal{T}_d \cup \{Male \sqsubseteq \neg Female;\ Male_{boy} \sqsubseteq \neg Female\} \cup \{P \sqsubseteq P \mid P \in \mathbf{P}\}$

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

  $\{S_1, S_2, S_3, S_4\}$ is complete!

Repair Answer Set Computation

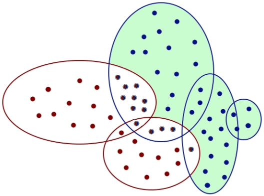
✓ Compute complete support families **S** for all DL-atoms of Π

- Construct $\hat{\Pi}$ from $\Pi = \langle \mathcal{O}, \mathcal{P} \rangle$:
  - Replace all DL-atoms a with normal atoms e_a
  - 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 $\mathcal{A}$: $S_{gr}^{\hat{I}} \leftarrow Gr(\mathbf{S}, \hat{I}, \mathcal{A})$

✓ Find $\mathcal{A}'$, such that

- ✓ For all $a \in D_p$: there is $S \in S_{gr}^{\hat{I}}(a)$, s.t. $S \cap \mathcal{A}' \neq \emptyset$ or $S \subseteq \mathcal{A}_a$
- ✓ For all $a' \in D_n$: for all $S \in S_{gr}^{\hat{I}}(a')$: $S \cap \mathcal{A}' = \emptyset$ and $S \not\subseteq \mathcal{A}_{a'}$
- ✓ Minimality check of $\hat{I}|_{\Pi}$ wrt. $\Pi' = \langle \mathcal{O}', \mathcal{P} \rangle$, $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$



Repair Answer Set Computation

Algorithm 1: $SupRAnsSet$: all deletion repair answer sets

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

(a) compute a complete set $\mathbf{S}$ of nongr. supp. sets for the DL-atoms in Π
(b) **for** $\hat{I} \in AS(\hat{\Pi})$ **do**
  $D_p \leftarrow \{a \mid e_a \in \hat{I}\}$; $D_n \in \{a \mid ne_a \in \hat{I}\}$; $\mathbf{S}^{\hat{I}}_{gr} \leftarrow Gr(\mathbf{S}, \hat{I}, \mathcal{A})$;
(c)   **if** $\mathbf{S}^{\hat{I}}_{gr}(a) \neq \emptyset$ *for* $a \in D_p$ *and every* $S \in \mathbf{S}^{\hat{I}}_{gr}(a)$ *for* $a \in D_n$ *fulfills* $S \cap \mathcal{A} \neq \emptyset$ **then**
(d)   **for** *all* $a \in D_p$ **do**
(e)   **if** *some* $S \in \mathbf{S}^{\hat{I}}_{gr}(a)$ *exists s.t.* $S \cap \mathcal{A} = \emptyset$ **then** pick next a
      else remove each S from $\mathbf{S}^{\hat{I}}_{gr}(a)$ s.t. $S \cap \mathcal{A} \cap \bigcup_{a' \in D_n} \mathbf{S}^{\hat{I}}_{gr}(a') \neq \emptyset$
(f)   **if** $\mathbf{S}^{\hat{I}}_{gr}(a) = \emptyset$ **then** pick next $\hat{I}$
    end
(g) $\mathcal{A}' \leftarrow \mathcal{A} \setminus \bigcup_{a' \in D_n} \mathbf{S}^{\hat{I}}_{gr}(a')$;
(h)   **if** $flpFND(\hat{I}, \langle \mathcal{T} \cup \mathcal{A}', \mathcal{P} \rangle)$ **then** output $\hat{I}|_{\Pi}$
  end
end

Repair Answer Set Computation

Algorithm 1: $SupRAnsSet$: all deletion repair answer sets

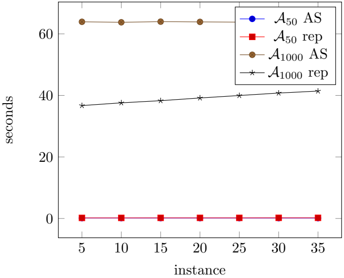
Input: $\Pi = \langle \mathcal{T} \cup \mathcal{A}, \mathcal{P} \rangle$
Output: $flpRAS(\Pi)$

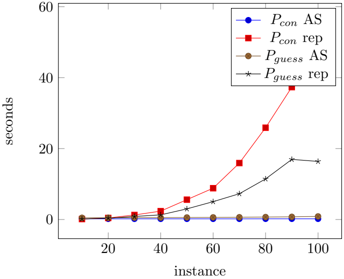
(a) compute a complete set $\mathbf{S}$ of nongr. supp. sets for the DL-atoms in Π
(b) **for** $\hat{I} \in AS(\hat{\Pi})$ **do**

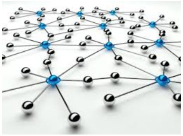
SupRAnsSet is sound and complete
wrt. deletion repair answer sets!

(e) **if** *some* $S \in \mathbf{S}^{\hat{I}}_{gr}(a)$ *exists s.t.* $S \cap \mathcal{A} = \emptyset$ **then** pick next a
else remove each S from $\mathbf{S}^{\hat{I}}_{gr}(a)$ s.t. $S \cap \mathcal{A} \cap \bigcup_{a' \in D_n} \mathbf{S}^{\hat{I}}_{gr}(a') \neq \emptyset$
(f) **if** $\mathbf{S}^{\hat{I}}_{gr}(a) = \emptyset$ **then** pick next $\hat{I}$
end
(g) $\mathcal{A}' \leftarrow \mathcal{A} \setminus \bigcup_{a' \in D_n} \mathbf{S}^{\hat{I}}_{gr}(a')$;
(h) **if** $flpFND(\hat{I}, \langle \mathcal{T} \cup \mathcal{A}', \mathcal{P} \rangle)$ **then** output $\hat{I}|_{\Pi}$
end
end

Experiments







Related Work

Inconsistenies 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 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*$_{\mathcal{A}}$ are small and efficiently computable
- Effective sound and complete algorithm *SupRAnsSet* 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. $\mathcal{EL}$)
- Computing preferred repairs (e.g. σ-selection [Eiter *et* al, *IJCAI'2013*])

References I

Meghyn Bienvenu.
On the complexity of consistent query answering in the presence of simple ontologies.
In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 705–711, Toronto, Ontario, Canada, July 2012. American Association for Artificial Intelligence.

Diego Calvanese, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati.
Tractable reasoning and efficient query answering in description logics: The DL-Lite family.
Journal of Automated Reasoning, 39(3):385–429, October 2007.

References II

Diego Calvanese, Magdalena Ortiz, Mantas Simkus, and Giorgio Stefanoni.
The complexity of explaining negative query answers in DL-Lite.
In *Proceedings of the 13th International Conference on the Principles of Knowledge Representation and Reasoning*, Rome, Italy, June 2012. American Association for Artificial Intelligence.

Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo.
Inconsistency-tolerant semantics for description logic ontologies.
In *Proceedings of the 19th Italian Symposium on Advanced Database Systems*, pages 103–117, Bressanone/Brixen, Italy, September 2010. Springer.

DL-program: syntax

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

DL-atom is of the form $DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](\mathbf{t})$, $m \geq 0$, where

- $S_i \in \mathbf{C} \cup \mathbf{R}$;
- $op_i \in \{\uplus, \cupdot, \cap\!\!\!\!-\}$;
- $p_i \in \mathcal{P}$ (unary or binary);
- $Q(\mathbf{t})$ is a DL-query:
    - $C(t_1)$, $\neg C(t_1)$, $\mathbf{t} = t_1$, where $C \in \mathbf{C}$;
    - $R(t_1, t_2)$, $\neg R(t_1, t_2)$, $\mathbf{t} = t_1, t_2$, where $R \in \mathbf{R}$.
    - $C \sqsubseteq D$, $C \not\sqsubseteq D$, $\mathbf{t} = \epsilon$, where $C, D \in \mathbf{C} \cup \{\top, \bot\}$;

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

$$a_1 \vee \ldots \vee a_n \leftarrow b_1, \ldots b_k, not\ b_{k+1}, \ldots, not\ b_m,$$

$m \geq k \geq 0$, a_i is a classical literal; b_i is a classical literal or a DL-atom.

DL-program: semantics

Consider grounding $grd(\Pi) = \langle \mathcal{O}, grd(P) \rangle$ 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 ℓ: $I \models^{\mathcal{O}} \ell$ iff $\ell \in I$;
- for ground DL-atom $a = DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](\mathbf{c})$:

  $$I \models^{\mathcal{O}} a$$

  iff $\tau(\langle \mathcal{T}, \mathcal{A} \cup \lambda^I(a) \rangle) \models Q(\mathbf{c})$, where $\tau(\mathcal{O})$ is a modular translation of $\mathcal{O}$ to FOL, $\lambda^I(a) = \bigcup_{i=1}^{m} A_i(I)$ is a DL-update of $\mathcal{O}$ under I by a:
  - $A_i(I) = \{S_i(t) \mid p_i(t) \in I\}$, for $op_i = \uplus$;
  - $A_i(I) = \{\neg S_i(t) \mid p_i(t) \in I\}$, for $op_i = \cupdot$;
  - $A_i(I) = \{\neg S_i(t) \mid p_i(t) \notin I\}$, for $\Cap$.

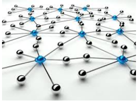
FLP-reduct $\rho_{flp} P^I$ of P is a set of ground DL-rules r s.t. $I \models b^+(r), \; I \not\models b^-(r)$.

Weak-reduct $\rho_{weak} P^I$ 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 $\rho_{flp} P^I$.

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

Network Benchmark

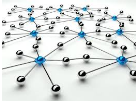
$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \exists \mathit{forbid} \sqsubseteq \mathit{Block} & (4)\ \mathit{edge}(n_i, n_j) \\ (2)\ \mathit{Broken} \sqsubseteq \mathit{Block} & (5)\ \ldots \\ (3)\ \mathit{Block} \sqsubseteq \neg \mathit{Avail} & (6)\ \ldots \end{array} \right\}$$



$$\mathcal{P}_{guess} = \left\{ \begin{array}{ll} (1) & go(X, Y) \leftarrow open(X), open(Y), \mathrm{DL}[;\, edge](X, Y). \\ (2) & route(X, Z) \leftarrow route(X, Y), route(Y, Z). \\ (3) & route(X, Y) \leftarrow not\ \mathrm{DL}[Block \uplus block;\, forbid](X, Y), go(X, Y). \\ (4) & open(X) \vee block(X) \leftarrow not\ \mathrm{DL}[;\, \neg Avail](X), node(X). \\ (5) & negIs(X) \leftarrow node(X), route(X, Y), X \neq Y. \\ (6) & \bot \leftarrow node(X), not\ negIs(X). \end{array} \right\}$$

Network Benchmark

$$\mathcal{O} = \left\{ \begin{array}{ll} (1)\ \exists \mathit{forbid} \sqsubseteq \mathit{Block} & (4)\ \mathit{edge}(n_i, n_j) \\ (2)\ \mathit{Broken} \sqsubseteq \mathit{Block} & (5)\ \dots \\ (3)\ \mathit{Block} \sqsubseteq \neg \mathit{Avail} & (6)\ \dots \end{array} \right\}$$



$$\mathcal{P}_{con} = \left\{ \begin{array}{l} (1)\ \ \mathit{go}(X,Y) \leftarrow \mathit{open}(X), \mathit{open}(Y), \mathrm{DL}[;\mathit{edge}](X,Y). \\ (2)\ \ \mathit{route}(X,Z) \leftarrow \mathit{route}(X,Y), \mathit{route}(Y,Z). \\ (3')\ \ \mathit{route}(X,Y) \leftarrow \mathit{go}(X,Y), \mathit{not}\ \mathrm{DL}[;\mathit{forbid}](X,Y). \\ (4')\ \ \mathit{open}(X) \leftarrow \mathit{node}(X), \mathit{not}\ \mathrm{DL}[;\neg \mathit{Avail}](X). \\ (5)\ \ \mathit{negIs}(X) \leftarrow \mathit{node}(X), \mathit{route}(X,Y), X \neq Y. \\ (6')\ \ \bot \leftarrow \mathit{in}(X), \mathit{out}(Y), \mathit{not}\ \mathit{route}(X,Y). \end{array} \right\}$$