Data Repair of Inconsistent DL-Programs

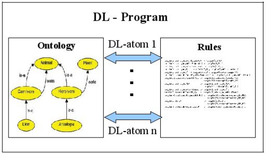
Thomas Eiter Michael Fink Daria Stepanova

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

IJCAI 2013 –August 6, 2013

Motivation

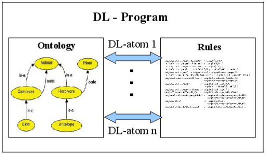
- DL-program: ontology + rules (loose coupling combination approach);
- DL-atoms serve as query interfaces to ontology;
- Possibility to add information from the rule part to ontology prior to querying it allows for bidirectional information flow.



However, information exchange between rules and ontology can have unforeseen effects and cause **inconsistency** of the DL-program (absence of answer sets).

Motivation

- DL-program: ontology + rules (loose coupling combination approach);
- DL-atoms serve as query interfaces to ontology;
- Possibility to add information from the rule part to ontology prior to querying it allows for bidirectional information flow.



However, information exchange between rules and ontology can have unforeseen effects and cause **inconsistency** of the DL-program (absence of answer sets).

In this work: Repair data part of the ontology ($DL\text{-}Lite_{\mathcal{A}}$), i.e. change ontology ABox s.t. the resulting DL-program is consistent.

Overview

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.

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

$DL\text{-}Lite_{\mathcal{A}}$

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

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

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

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

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

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

Example

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

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.

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

Example

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

Conjunctive query answering in *DL-Lite*$_{\mathcal{A}}$ is tractable [Calvanese *et al.*, 2007].

Example: DL-program

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

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

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

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

- interpretation: $I = \{\mathit{ischildof}(\mathit{john}, \mathit{alex}), \mathit{boy}(\mathit{john}), \mathit{hasfather}(\mathit{john}, \mathit{pat})\}$;
- satisfaction relation: $I \models^{\mathcal{O}} \mathit{boy}(\mathit{john})$; $I \models^{\mathcal{O}} \mathit{DL}[; \mathit{hasParent}](\mathit{john}, \mathit{pat})$;
- semantics is given in terms of answer sets, which are x-founded models;
- *flp* and *weak* semantics are relevant in this work;
- I is both *weak*- and *flp*-founded model.

Example: Inconsistent DL-program

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

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

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

Example: Inconsistent DL-program

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

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

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

Example: Inconsistent DL-program

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

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

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

Example: Inconsistent DL-program

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

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

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

No answer sets.

Example: Inconsistent DL-program

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

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

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

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

Example: Inconsistent DL-program

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

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

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

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

Repair Answer Sets

Definition

Let $\Pi = \langle \mathcal{O}, P \rangle$, $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-program,

- an ABox $\mathcal{A}'$ is an *x-repair* of Π if
  - $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ is consistent;
  - $\Pi' = \langle \mathcal{O}', P \rangle$ has some x-answer set.

$rep_x(\Pi)$ is the set of all x-repairs of Π.

- I is an *x-repair answer set* of Π, if $I \in AS_x(\Pi')$, where $\Pi' = \langle \mathcal{O}', P \rangle, \mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$, and $\mathcal{A}' \in rep_x(\Pi)$.

$RAS_x(\Pi)$ is the set of all x-repair AS of Π.

$rep^I_x(\Pi)$ is the set of all $\mathcal{A}'$ under which I is an x-repair answer set of Π.

Repair Answer Sets

Definition

Let $\Pi = \langle \mathcal{O}, P \rangle$, $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-program,

- an ABox $\mathcal{A}'$ is an *x-repair* of Π if
  - $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ is consistent;
  - $\Pi' = \langle \mathcal{O}', P \rangle$ has some x-answer set.

$rep_x(\Pi)$ is the set of all x-repairs of Π.

- I is an *x-repair answer set* of Π, if $I \in AS_x(\Pi')$, where $\Pi' = \langle \mathcal{O}', P \rangle, \mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$, and $\mathcal{A}' \in rep_x(\Pi)$.

$RAS_x(\Pi)$ is the set of all x-repair AS of Π.

$rep_x^I(\Pi)$ is the set of all $\mathcal{A}'$ under which I is an x-repair answer set of Π.

Example

$I_1 = \{ischildof(john, alex), boy(john)\}$ is an *flp*-repair answer set with repair $\mathcal{A}'_1 = \{Male(pat), Male(john)\}$; $\mathcal{A}'_1 \in rep^I_{flp}(\Pi)$.

Complexity of Repair Answer Sets

Theorem

Deciding $AS_x(\Pi) \neq \emptyset$ and deciding $RAS_x(\Pi) \neq \emptyset$ have in all cases the same complexity.

Π	$RAS_{FLP}(\Pi) \neq \emptyset$	$RAS_{weak}(\Pi) \neq \emptyset$
normal	Σ_2^P-complete	NP-complete
disjunctive	Σ_2^P-complete	Σ_2^P-complete

Membership:

- guess repair $\mathcal{A}'$ together with I and proceed with the check as usual;
- deciding $I \models^{\mathcal{O}} a$ is feasible in polynomial time if $\mathcal{O}$ is in $DL\text{-}Lite_{\mathcal{A}}$;

Hardness: for normal FLP AS hardness proof of ordinary disjunctive LP can be adapted, for other cases hardness is inherited from ordinary ASP.

DL-program Evaluation

Algorithm 1: $AnsSet$: Compute $AS_x(\Pi)$

Input: A DL-program Π, $x \in \{weak, flp\}$
Output: $AS_x(\Pi)$
for $\hat{I} \in AS(\hat{\Pi})$ **do**
  if $CMP(\hat{I}, \Pi) \wedge xFND(\hat{I}, \Pi)$ **then**
    output $\hat{I}|_{\Pi}$
  end
end

- $\hat{\Pi}$ is Π with all DL-atoms a substituted by ordinary atoms e_a plus additional guess rules for values of e_a;
- $CMP(\hat{I}, \Pi)$ is a compatibility check, i.e. check whether the values of DL-atoms coincide with the values of their replacement atoms in $\hat{I}$;
- $xFND(\hat{I}, \Pi)$ is x-foundedness check;
- $\hat{I}|_{\Pi}$ is a restriction of $\hat{I}$ to original language of Π.

DL-program Evaluation

Algorithm 1: $AnsSet$: Compute $AS_x(\Pi)$

Input: A DL-program Π, $x \in \{weak, flp\}$
Output: $AS_x(\Pi)$

```
(1)     for Î ∈ AS(Π̂) do
(2a,b)      if CMP(Î, Π) ∧ xFND(Î, Π) then
                output Î|Π
            end
        end
```

Reasons for inconsistency:

1. $\hat{\Pi}$ does not have any answer sets;

2. for all $\hat{I} \in AS(\Pi)$:

- **a.** compatibility check failed or
- **b.** *x*-foundedness check failed.

Ontology Repair Problem

To address the compatibility check issue we introduce:

Definition

A ontology repair problem (ORP) is a triple $\mathcal{P} = \langle \mathcal{O}, D_1, D_2 \rangle$, where $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an ontology and $D_i = \{\langle U^i_j, Q^i_j \rangle | 1 \leq j \leq m_i\},\ i = 1, 2$ are sets of pairs where U^i_j is any ABox and each Q^i_j is a DL-query.

A repair (solution) for $\mathcal{P}$ is any ABox $\mathcal{A}'$ s.t.

- $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ is consistent;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U^1_k \rangle) \models Q^1_j$ holds for $1 \leq j \leq m_1$;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U^2_k \rangle) \not\models Q^2_j$ holds for $1 \leq j \leq m_2$.

Ontology Repair Problem

To address the compatibility check issue we introduce:

Definition

A ontology repair problem (ORP) is a triple $\mathcal{P} = \langle \mathcal{O}, D_1, D_2 \rangle$, where $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an ontology and $D_i = \{\langle U_j^i, Q_j^i \rangle | 1 \leq j \leq m_i\},\ i = 1, 2$ are sets of pairs where U_j^i is any ABox and each Q_j^i is a DL-query.

Example

$$\Pi = \langle \mathcal{O}, P \rangle, \text{ where } P = \left\{ \begin{array}{l} p(c);\ \ r(c);\ \ q(c) \leftarrow \underbrace{DL[C \uplus r; D](c)}_{a_1}; \\ \bot \leftarrow \underbrace{DL[D \uplus p, E \uplus r; \neg C](c)}_{a_2} \end{array} \right\}.$$

- $\hat{I} = \{p(c), r(c), q(c), e_{a_1}\}$: a_1 is guessed true, a_2 is guessed false;
- $\mathcal{P} = \langle \mathcal{O}, D_1, D_2 \rangle$, where
  - $D_1 = \{\langle \{\neg C(c)\}; D(c) \rangle\}$;
  - $D_2 = \{\langle \{D(c), \neg E(c)\}; \neg C(c) \rangle\}$.

Ontology Repair Problem

A repair (solution) for $\mathcal{P}$ is any ABox $\mathcal{A}'$ s.t.

- $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ is consistent;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U_k^1 \rangle) \models Q_j^1$ holds for $1 \leq j \leq m_1$;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U_k^2 \rangle) \not\models Q_j^2$ holds for $1 \leq j \leq m_2$.

Example

Let $\mathcal{O} = \langle \overbrace{E \sqsubseteq D, A \sqsubseteq D}^{\mathcal{T}}, \overbrace{\neg C(c)}^{\mathcal{A}} \rangle$;

- $\mathcal{P} = \langle \mathcal{O}, D_1, D_2 \rangle$, where
  - $D_1 = \{\langle \{\neg C(c)\}; D(c) \rangle\}$;
  - $D_2 = \{\langle \{D(c), \neg E(c)\}; \neg C(c) \rangle\}$.

Ontology Repair Problem

A repair (solution) for $\mathcal{P}$ is any ABox $\mathcal{A}'$ s.t.

- $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ is consistent;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U_k^1 \rangle) \models Q_j^1$ holds for $1 \leq j \leq m_1$;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U_k^2 \rangle) \not\models Q_j^2$ holds for $1 \leq j \leq m_2$.

Example

Let $\mathcal{O} = \langle \overbrace{E \sqsubseteq D, A \sqsubseteq D}^{\mathcal{T}}, \overbrace{\neg C(c)}^{\mathcal{A}} \rangle$;

- $\mathcal{P} = \langle \mathcal{O}, D_1, D_2 \rangle$, where
  - $D_1 = \{\langle \{\neg C(c)\}; D(c) \rangle\}$;
  - $D_2 = \{\langle \{D(c), \neg E(c)\}; \neg C(c) \rangle\}$.

$\mathcal{A}' = \{A(c)\}$ is a solution for $\mathcal{P}$.

Ontology Repair Problem

A repair (solution) for $\mathcal{P}$ is any ABox $\mathcal{A}'$ s.t.

- $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ is consistent;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U_k^1 \rangle) \models Q_j^1$ holds for $1 \leq j \leq m_1$;
- $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U_k^2 \rangle) \not\models Q_j^2$ holds for $1 \leq j \leq m_2$.

Example

Let $\mathcal{O} = \langle \overbrace{E \sqsubseteq D, A \sqsubseteq D}^{\mathcal{T}}, \overbrace{\neg C(c)}^{\mathcal{A}} \rangle$;

- $\mathcal{P} = \langle \mathcal{O}, D_1, D_2 \rangle$, where
  - $D_1 = \{\langle \{\neg C(c)\}; D(c) \rangle\}$;
  - $D_2 = \{\langle \{D(c), \neg E(c)\}; \neg C(c) \rangle\}$.

$\mathcal{A}' = \{A(c)\}$ is a solution for $\mathcal{P}$.

ORP is *NP*-complete in general, even if $\mathcal{O} = \emptyset$.

Selection Preferences

Consider a set $\mathcal{AB}$ of all possible ABoxes.
Function $\sigma : 2^{\mathcal{AB}} x \mathcal{AB} \rightarrow 2^{\mathcal{AB}}$ is a selection function.
$\sigma(S, \mathcal{A}) \subseteq S$ is a set of preferred ABoxes.

A selection $\sigma : 2^{\mathcal{AB}} x \mathcal{AB} \rightarrow 2^{\mathcal{AB}}$ is independent if
$\sigma(S, \mathcal{A}) = \sigma(S', \mathcal{A}) \cup \sigma(S \backslash S', \mathcal{A})$, whenever $S' \subseteq S$.

Example

- deletion repair is independent;
- set-minimal change repair is not independent;
- cardinality minimal change repair is not independent.

Tractable Cases of ORP

C1. bounded $\delta^{\pm}$-change: $\sigma_{\delta^{\pm},k}(S, \mathcal{A}) = \{\mathcal{A}' \mid |\mathcal{A}' \Delta \mathcal{A}| \leq k\}$, for some k;

C2. deletion repair: $\sigma_{del}(S, \mathcal{A}) = \{\mathcal{A}' \mid \mathcal{A}' \subseteq \mathcal{A}\}$;

C3. deletion δ^{+}: first apply σ_{del} and get $\mu(\mathcal{O})$ s.t. for all $1 \leq j \leq m_2$ $\tau(\langle \mathcal{T}, \mathcal{A}' \cup U_j^2 \rangle) \not\models Q_j^2$, then further compute $\sigma_{\delta^+}(S, \mu(\mathcal{O}))$;

C4. addition under bounded opposite polarity: $\sigma_{bop}(S, \mathcal{A}) = \{\mathcal{A}' \supseteq \mu(\mathcal{O}) || \mathcal{A}'^{+} \backslash \mathcal{A}| \leq k \text{ or } |\mathcal{A}'^{-} \backslash \mathcal{A}| \leq k\}$

C1 - C4 are independent.

Applicability of results for independent selections:

- deciding whether repair $\mathcal{A}'$ is selected by σ does not require looking at other repairs;
- without major complexity increase σs can be combined with
  - DB-style factorization and localization techniques;
  - local search.

Repair Answer Set Computation

Algorithm 2: $RepAns$: Compute $rep^{\hat{I}|_\Pi}_{(\sigma,x)}(\Pi)$

Input: $\Pi = \langle \mathcal{O}, P \rangle$, $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, $\hat{I} \in AS(\hat{\Pi})$, σ, $x \in \{weak, flp\}$
Output: $rep^{\hat{I}|_\Pi}_{(\sigma,x)}(\Pi)$

for $\mathcal{A}' \in ORP(\hat{I}, \Pi, \sigma)$ **do**
  if $CMP(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle) \wedge xFND(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle)$ **then**
    output $\mathcal{A}'$
  end
end

- *ORP*$(\hat{I}, \Pi, \sigma)$ computes σ repairs for $\hat{I}, \Pi$;
- *CMP*$(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle)$ checks whether $\hat{I}$ is compatible w.r.t. Π';
- *xFND*$(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle)$ checks whether $\hat{I}$ is x-founded w.r.t. Π'.

RepAnsSet outputs $\hat{I}$ if the result of *RepAns* is nonempty.

Repair Answer Set Computation

Algorithm 2: $RepAns$: Compute $rep^{\hat{I}|_{\Pi}}_{(\sigma,x)}(\Pi)$

Input: $\Pi = \langle \mathcal{O}, P \rangle$, $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, $\hat{I} \in AS(\hat{\Pi})$, σ, $x \in \{weak, flp\}$
Output: $rep^{\hat{I}|_{\Pi}}_{(\sigma,x)}(\Pi)$
for $\mathcal{A}' \in ORP(\hat{I}, \Pi, \sigma)$ **do**
  if $CMP(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle) \wedge xFND(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle)$ **then**
    output $\mathcal{A}'$
  end
end

- $ORP(\hat{I}, \Pi, \sigma)$ computes σ repairs for $\hat{I}, \Pi$;
- $CMP(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle)$ checks whether $\hat{I}$ is compatible w.r.t. Π';
- $xFND(\hat{I}, \langle \mathcal{T}, \mathcal{A}', P \rangle)$ checks whether $\hat{I}$ is x-founded w.r.t. Π'.

RepAnsSet outputs $\hat{I}$ if the result of *RepAns* is nonempty.

RepAns and *RepAnsSet* are sound and complete for independent σ.

Related Work

- Repairing ontologies
  - consistent query answering over *DL-Lite* ontologies based on repair technique [Lembo *et al.*, 2010], [Bienvenu, 2012];
  - QA to $DL\text{-}Lite_{\mathcal{A}}$ ontologies that miss expected tuples (abductive explanations corresponding to repairs) [Calvanese *et al.*, 2012].

- Repairing nonmonotonic logic programs
  - extended abduction for deleting minimal sets of rules (in reality addition is also possible) [Sakama and Inoue, 2003].
- Repairing inconsistent combination of rules and ontologies
  - paraconsistent semantics, based on the HT logic [Fink, 2012];
  - inconsistency tolerance in DL-programs [Pührer *et al.*, 2010].

Conclusion and Future Work

Conclusions:

- consideration of repair answer sets (RAS);
- same complexity as ordinary AS (for $\mathcal{O}$ in *DL-Lite*$_{\mathcal{A}}$);
- RAS computation by extending the existing evaluation algorithm;
- involvement of a generalized ontology repair problem (ORP);
- tractable cases for independent selections.

Future work:

- extending the work to other DLs ($\mathcal{EL}$++, RL);
- DL-programs with richer queries (unions of conjunctive queries);
- further σ-selections;
- optimization and implementation.

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.

- Michael Fink.
  Paraconsistent hybrid theories.
  In *Principles of Knowledge Representation and Reasoning: Proceedings of the 13th International Conference, KR*, pages 141–151, Rome, Italy, June 2012. American Association for Artificial Intelligence.

References III

- 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.
- Jörg Pührer, Stijn Heymans, and Thomas Eiter.
  Dealing with inconsistency when combining ontologies and rules using DL-programs.
  In *Proceedings of 7th Extended Semantic Web Conference, part I*, pages 183–197, Heraklion, Crete, Greece, May-June 2010. Springer.
- Chiaki Sakama and Katsumi Inoue.
  An abductive framework for computing knowledge base updates.
  Theory and Practice of Logic Programming, 3(6):671–713, May 2003.

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 $op_i = \cup\!\!\!-$;
- $A_i(I) = \{\neg S_i(t) \mid p_i(t) \notin I\}$, 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.