The Combined Approach
to Query Answering in DL-Lite

Roman Kontchakov\textsuperscript{1}, Carsten Lutz\textsuperscript{2}, David Toman\textsuperscript{3},
Frank Wolter\textsuperscript{4}, and Michael Zakharyaschev\textsuperscript{1}

\textsuperscript{1}Department of CS and Information Systems, Birkbeck College London, UK
\{roman,michael\}@dcs.bbk.ac.uk

\textsuperscript{2}Fachbereich Mathematik und Informatik, Universität Bremen, Germany
clu@informatik.uni-bremen.de

\textsuperscript{3}D.R. Cheriton School of Computer Science, University of Waterloo, Canada
david@cs.uwaterloo.ca

\textsuperscript{4}Department of Computer Science, University of Liverpool, UK
frank@csc.liv.ac.uk
Queries and Ontologies

Ontology-based Data Access

Enriches (query answers over) *explicitly represented data* using *background knowledge* (captured using an *ontology*.)

Problem: answering queries is *EXPENSIVE* (data complexity)

⇒ *large data sets* and *(relatively) large ontologies.*

⇒ *need for lightweight ontology* and *query languages;*

*DL-Lite* (family) and *conjunctive queries.*

... introduced by [Calvanese et al.]
Queries and Ontologies

Ontology-based Data Access

Enriches (query answers over) *explicitly represented data* using *background knowledge* (captured using an *ontology*).

Example

- Bob is a BOSS  
  *(explicit data)*
- Every BOSS is an EMPloyee  
  *(ontology)*

**List all EMPloyees** $\Rightarrow \{\text{Bob}\}$  
*(query)*

Problem: answering queries is *EXPENSIVE* (data complexity)  
$\Rightarrow$ *large data sets* and *(relatively)* *large ontologies*.  
$\Rightarrow$ need for lightweight *ontology* and *query languages*;

*DL-Lite (family)* and *conjunctive queries*.

Konchatov et al. ()  
KR 2008 2 / 11
Queries and Ontologies

Ontology-based Data Access

Enriches (query answers over) explicitly represented data using background knowledge (captured using an ontology.)

Problem: answering queries is EXPENSIVE (data complexity)
⇒ large data sets and (relatively) large ontologies.
⇒ need for lightweight ontology and query languages;

DL-Lite (family) and conjunctive queries.

... introduced by [Calvanese et al.]
Queries and Ontologies

Ontology-based Data Access

Enriches (query answers over) *explicitly represented data* using *background knowledge* (captured using an *ontology*.)

Problem: answering queries is *EXPENSIVE* (data complexity)

⇒ *large data sets* and *(relatively) large ontologies*.
⇒ need for *lightweight ontology* and *query languages*;

*DL-Lite* (family) and *conjunctive queries*.

... introduced by [Calvanese et al.]
Approaches to Ontology-based Data Access

Main Task

INPUT: Ontology \((\mathcal{T})\), Data \((\mathcal{A})\), and a Query \((Q)\)

Knowledge Base \((\mathcal{K})\)

OUTPUT: \(\{\ a \mid \mathcal{K} \models Q[a]\}\)

Approaches:

1. Reduction to \textit{standard reasoning} (e.g., satisfiability)
2. Reduction to \textit{querying a relational database}
   \(\Rightarrow\) very good at \(\{\ a \mid \mathcal{A} \models Q[a]\}\) for range restricted \(Q\)

\(\ldots\) what do we do with \(\mathcal{T}\)?
Approaches to Ontology-based Data Access

Main Task

INPUT: Ontology \((\mathcal{T})\), Data \((\mathcal{A})\), and a Query \((\mathcal{Q})\) and Knowledge Base \((\mathcal{K})\)

OUTPUT: \(\{a \mid \mathcal{K} \models \mathcal{Q}[a]\}\)

Approaches:

1. Reduction to standard reasoning (e.g., satisfiability)
2. Reduction to querying a relational database
   \[\Rightarrow\] very good at \(\{a \mid \mathcal{A} \models \mathcal{Q}[a]\}\) for range restricted \(\mathcal{Q}\)

\[\ldots\text{what do we do with }\mathcal{T}\?\]
Definitions & Background

**Definition (DL-Lite$^\lor_{horn}$)**

roles: \( R ::= P \mid P^\neg, \)  

concepts: \( C ::= \bot \mid A \mid \geq m R. \)

where \( P \in N_R, A \in N_C \) and \( m > 0. \)

1. An ontology (TBox) is a finite set \( T \) of concept inclusions \( C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C; \)
2. The Data (ABox) is a finite set \( A \) of concept and role assertions \( C(a) \) and \( R(a, b); \)
3. A Conjunctive Query (CQ):
   an existentially quantified finite conjunction of atoms.
The Master Plan

IDEA:

1. Incorporate the background knowledge (i.e., $\mathcal{T}$) into the data.
   ⇒ make implicit knowledge explicit (data completion).

2. Use the data completion (only) to answer queries
   ⇒ and use a relational system to do this efficiently.

Issues:

1. How to complete the data?
   Naive unfolding of $\mathcal{T}$: large/infinite (due to existentials)
   ⇒ we define a canonical interpretation $\mathcal{I}_K$ (representatives)

2. Can we then use the original Conjunctive Query?
   Not directly: $Q(\mathcal{I}_K)$ can produce “spurious matches”
   ⇒ we eliminate the spurious matches by rewriting the query
   (independently of $\mathcal{T}$ and $\mathcal{A}$)
The Master Plan

IDEA:

1. Incorporate the background knowledge (i.e., \( T \)) into the data.
   ⇒ make implicit knowledge explicit (data completion).
2. Use the data completion (only) to answer queries
   ⇒ and use a relational system to do this efficiently.

Example

\[ T = \{ BOSS \sqsubseteq EMP \}, \quad A = \{ BOSS(Bob) \}, \quad Q \equiv EMP(x) \]

1. \( I_K = \{ BOSS(Bob), EMP(Bob) \} \) (data completion)
2. \( Q(I_K) = \{ Bob \} \) (relational query)

Issues:

1. How to complete the data?
   - Naive unfolding of \( T \): large/infinite (due to existentials)
     ⇒ we define a canonical interpretation \( I_K \) (representatives)
2. Can we then use the original Conjunctive Query?
   - Not directly: \( Q(I_K) \) can produce "spurious matches"
     ⇒ we eliminate the spurious matches by rewriting the query (independently of \( T \) and \( A \))
The Master Plan

IDEA:

1. Incorporate the background knowledge (i.e., $\mathcal{T}$) into the data.
   ⇒ make implicit knowledge explicit (data completion).

2. Use the data completion (only) to answer queries
   ⇒ and use a relational system to do this efficiently.

Issues:

1. How to complete the data?
   Naive unfolding of $\mathcal{T}$: large/infinite (due to existentials)
   ⇒ we define a canonical interpretation $\mathcal{I}_K$ (representatives)

2. Can we then use the original Conjunctive Query?
   Not directly: $Q(\mathcal{I}_K)$ can produce “spurious matches”
   ⇒ we eliminate the spurious matches by rewriting the query
     (independently of $\mathcal{T}$ and $\mathcal{A}$)
Canonical Interpretations

**ABox completion: the Canonical Interpretation $\mathcal{I}_\mathcal{K}$**

\[
\begin{align*}
A^{\mathcal{I}_\mathcal{K}} &= \{ a \in \text{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a) \} \cup \{ c_R \in \Delta^{\mathcal{I}_\mathcal{K}} \mid T \models \exists R^{-} \subseteq A \}, \\
P^{\mathcal{I}_\mathcal{K}} &= \{ (a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A} \} \cup \\
&\quad \{ (d, c_P) \in \Delta^{\mathcal{I}_\mathcal{K}} \times N^{T}_I \mid d \sim c_P \} \cup \{ (c_{P^{-}}, d) \in N^{T}_I \times \Delta^{\mathcal{I}_\mathcal{K}} \mid d \sim c_{P^{-}} \}
\end{align*}
\]

...$c_R$’s only used “when necessary” (for generating roles)

**Lemma**

There are queries

- $q^T_A$ s.t. $\text{ans}(q^T_A, \mathcal{A}) = A^{\mathcal{I}_\mathcal{K}}$, and
- $q^T_P$ s.t. $\text{ans}(q^T_P, \mathcal{A}) = P^{\mathcal{I}_\mathcal{K}}$

for every KB $(T, \mathcal{A})$ and primitive concept $A$ and role $P$. 

Free consistency test: $q^T_\perp(\mathcal{A}) = \emptyset$
Canonical Interpretations

ABox completion: the Canonical Interpretation $\mathcal{I}_K$

$A^{\mathcal{I}_K} = \{a \in \text{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a)\} \cup \{c_R \in \Delta^{\mathcal{I}_K} \mid \mathcal{T} \models \exists R^{-} \subseteq A\}$,

$P^{\mathcal{I}_K} = \{(a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A}\} \cup \{(d, c_P) \in \Delta^{\mathcal{I}_K} \times \mathcal{N}^T \mid d \rightsquigarrow c_P\} \cup \{(c_{P^{-}}, d) \in \mathcal{N}^T \times \Delta^{\mathcal{I}_K} \mid d \rightsquigarrow c_{P^{-}}\}$

... $c_R$’s only used “when necessary” (for generating roles)

Lemma

There are queries

- $q^T_A$ s.t. $\text{ans}(q^T_A, \mathcal{A}) = A^{\mathcal{I}_K}$, and
- $q^T_P$ s.t. $\text{ans}(q^T_P, \mathcal{A}) = P^{\mathcal{I}_K}$

for every KB $(\mathcal{T}, \mathcal{A})$ and primitive concept $A$ and role $P$.

Free consistency test: $q^T_{\perp}(\mathcal{A}) = \emptyset$
Canonical Interpretations

### ABox completion: the Canonical Interpretation $\mathcal{I}_K$

$$\begin{align*}
A^{\mathcal{I}_K} & = \{ a \in \text{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a) \} \cup \{ c_R \in \Delta^{\mathcal{I}_K} \mid \mathcal{T} \models \exists R^\bot \subseteq A \}, \\
P^{\mathcal{I}_K} & = \{ (a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A} \} \cup \\
& \quad \{ (d, c_P) \in \Delta^{\mathcal{I}_K} \times N_T^I \mid d \leadsto c_P \} \cup \{ (c_{P^-}, d) \in N_T^I \times \Delta^{\mathcal{I}_K} \mid d \leadsto c_{P^-} \}
\end{align*}$$

... $c_R$’s only used “when necessary” (for generating roles)

#### Lemma

There are queries

- $q^T_A$ s.t. $\text{ans}(q^T_A, \mathcal{A}) = A^{\mathcal{I}_K}$, and
- $q^T_P$ s.t. $\text{ans}(q^T_P, \mathcal{A}) = P^{\mathcal{I}_K}$

for every KB $(\mathcal{T}, \mathcal{A})$ and primitive concept $A$ and role $P$. 

Free consistency test: $q^T_U(\mathcal{A}) = \emptyset$
Canonical Interpretations

ABox completion: the Canonical Interpretation $\mathcal{I}_K$

$A^{\mathcal{I}_K} = \{ a \in \text{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a) \} \cup \{ c_R \in \Delta^{\mathcal{I}_K} \mid \mathcal{T} \models \exists R^- \sqsubseteq A \}$,

$P^{\mathcal{I}_K} = \{ (a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A} \} \cup \{(d, c_P) \in \Delta^{\mathcal{I}_K} \times \text{NT}^T \mid d \sim c_P \} \cup \{(c_{P^-}, d) \in \text{NT}^T \times \Delta^{\mathcal{I}_K} \mid d \sim c_{P^-} \}$

...$c_R$'s only used “when necessary” (for generating roles)

Example

$\mathcal{T} = \{ \text{EMP} \sqsubseteq \exists \text{MANAGES}, \exists \text{MANAGES}^- \sqsubseteq \text{BOSS}, \text{BOSS} \sqsubseteq \text{EMP} \}$

$\mathcal{A} = \{ \text{EMP}(Bob), \text{EMP}(Sue) \}$

Then $\text{EMP}^{\mathcal{I}_K} = \{ Bob, Sue, c_M \}$, $\text{BOSS}^{\mathcal{I}_K} = \{ c_M \}$, and $\text{MANAGES}^{\mathcal{I}_K} = \{ (Bob, c_M), (Sue, c_M), (c_M, c_M) \}$. $\mathcal{I}_K$ is NOT model of $(\mathcal{T}, \mathcal{A})$ in general.
Canonical Interpretations

**ABox completion: the Canonical Interpretation** $\mathcal{I}_K$

Let $K$ be a knowledge base.

- **$A^{\mathcal{I}_K}$**
  
  $A^{\mathcal{I}_K} = \{ \mathcal{I} \models A(a) | a \in \text{Ind}(\mathcal{A}) \} \cup \{ c_R \in \Delta^{\mathcal{I}_K} | \mathcal{T} \models \exists R^- \sqsubseteq A \}$,

- **$P^{\mathcal{I}_K}$**
  
  $P^{\mathcal{I}_K} = \{ (a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) | P(a, b) \in \mathcal{A} \} \cup \{ (d, c_P) \in \Delta^{\mathcal{I}_K} \times N^{\mathcal{T}} | d \sim c_P \} \cup \{ (c_{P^-}, d) \in N^{\mathcal{T}} \times \Delta^{\mathcal{I}_K} | d \sim c_{P^-} \}$

  ... $c_R$'s only used “when necessary” (for generating roles)

**Example**

$\mathcal{T} = \{ \text{EMP} \sqsubseteq \exists \text{MANAGES}, \exists \text{MANAGES}^- \sqsubseteq \text{BOSS}, \text{BOSS} \sqsubseteq \text{EMP} \}$

$\mathcal{A} = \{ \text{EMP}(\text{Bob}), \text{EMP}(\text{Sue}) \}$

Then $\text{EMP}^{\mathcal{I}_K} = \{ \text{Bob}, \text{Sue}, c_M \}$, $\text{BOSS}^{\mathcal{I}_K} = \{ c_M \}$, and $\text{MANAGES}^{\mathcal{I}_K} = \{ (\text{Bob}, c_M), (\text{Sue}, c_M), (c_M, c_M) \}$.

$\mathcal{I}_K$ is NOT model of $(\mathcal{T}, \mathcal{A})$ in general.
Canonical Interpretations

ABox completion: the Canonical Interpretation $I_{\mathcal{K}}$

\[ A_{I_{\mathcal{K}}} = \{ a \in \text{Ind}(\mathcal{A}) \mid \mathcal{K} \models A(a) \} \cup \{ c_R \in \Delta_{I_{\mathcal{K}}} \mid \mathcal{T} \models \exists R^- \subseteq A \}, \]

\[ P_{I_{\mathcal{K}}} = \{ (a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A} \} \cup \{ (d, c_P) \in \Delta_{I_{\mathcal{K}}} \times N_{\mathcal{T}} \mid d \leadsto c_P \} \cup \{ (c_{P^-}, d) \in N_{\mathcal{T}} \times \Delta_{I_{\mathcal{K}}} \mid d \leadsto c_{P^-} \} \]

\[ \ldots c_R \text{'s only used "when necessary" (for generating roles)} \]

Lemma

There are queries

- $q^T_A$ s.t. $\text{ans}(q^T_A, \mathcal{A}) = A_{I_{\mathcal{K}}}$, and
- $q^T_P$ s.t. $\text{ans}(q^T_P, \mathcal{A}) = P_{I_{\mathcal{K}}}$

for every KB $(\mathcal{T}, \mathcal{A})$ and primitive concept $A$ and role $P$.

free consistency test: $q^T_{I_{\mathcal{K}}} (\mathcal{A}) = \emptyset$
Canonical Interpretations

ABox completion: the Canonical Interpretation $\mathcal{I}_K$

$A^{\mathcal{I}_K} = \{ a \in \text{Ind}(\mathcal{A}) \mid K \models A(a) \} \cup \{ c_R \in \Delta^{\mathcal{I}_K} \mid \mathcal{T} \models \exists R^- \subseteq A \}$,

$P^{\mathcal{I}_K} = \{ (a, b) \in \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mid P(a, b) \in \mathcal{A} \} \cup \{ (d, c_P) \in \Delta^{\mathcal{I}_K} \times N^T_1 \mid d \leadsto c_P \} \cup \{ (c_P^-, d) \in N^T_1 \times \Delta^{\mathcal{I}_K} \mid d \leadsto c_P^- \}$

... $c_R$’s only used “when necessary” (for generating roles)

Lemma

There are queries

- $q^T_A$ s.t. $\text{ans}(q^T_A, \mathcal{A}) = A^{\mathcal{I}_K}$, and
- $q^T_P$ s.t. $\text{ans}(q^T_P, \mathcal{A}) = P^{\mathcal{I}_K}$

for every KB $(\mathcal{T}, \mathcal{A})$ and primitive concept $A$ and role $P$.

free consistency test: $q^T_{\bot}(\mathcal{A}) = \emptyset$
Query Rewriting

Example

\[ T = \{ \text{EMP} \sqsubseteq \exists \text{MANAGES}, \exists \text{MANAGES}^- \sqsubseteq \text{BOSS}, \text{BOSS} \sqsubseteq \text{EMP} \} \]

\[ A = \{ \text{EMP}(Bob), \text{EMP}(Sue) \} \]

Queries:

1. \( \exists v. \text{MANAGES}(v, v) \)
2. \( \exists y. \text{MANAGES}(x, y) \land \text{MANAGES}(z, y) \)

Query Rewriting

\[ \exists \bar{u}. \varphi \mapsto \exists \bar{u}. \varphi \land \varphi_1 \land \varphi_2 \land \varphi_3 \]

where

- \( \varphi_1 \) eliminates answers containing \( c_R \)'s;
- \( \varphi_2 \) eliminates problem (1) above; and
- \( \varphi_3 \) eliminates problem (2) above.

selections in SQL
Query Rewriting

Example

\[ T = \{ EMP \sqsubseteq \exists \text{MANAGES}, \exists \text{MANAGES}^{-} \sqsubseteq BOSS, BOSS \sqsubseteq EMP \} \]

\[ A = \{ EMP(Bob), EMP(Sue) \} \]

Queries:

1. \( \exists v. \text{MANAGES}(v, v) \)
2. \( \exists y. \text{MANAGES}(x, y) \land \text{MANAGES}(z, y) \)

\[ \mathcal{I}_K : \]

- Bob
- Sue
- \( c_M \)

\[ Q_1(\mathcal{I}_K) = \{ c_M \} \]

\[ Q_2(\mathcal{I}_K) = \{ (Bob, Sue) \} \]
Query Rewriting

Example

\[ T = \{ \text{EMP} \sqsubseteq \exists \text{MANAGES}, \exists \text{MANAGES}^- \sqsubseteq \text{BOSS}, \text{BOSS} \sqsubseteq \text{EMP} \} \]

\[ A = \{ \text{EMP}(Bob), \text{EMP}(Sue) \} \]

Queries:

1. \( \exists v. \text{MANAGES}(v, v) \)
2. \( \exists y. \text{MANAGES}(x, y) \land \text{MANAGES}(z, y) \)

Query Rewriting

\[ \exists \bar{u}. \varphi \quad \mapsto \quad \exists \bar{u}. \varphi \land \varphi_1 \land \varphi_2 \land \varphi_3 \]

where

- \( \varphi_1 \) eliminates answers containing \( c_R \)'s;
- \( \varphi_2 \) eliminates problem (1) above; and
- \( \varphi_3 \) eliminates problem (2) above.

\begin{align*}
\text{selections} \quad \text{in SQL}
\end{align*}
UNA or not UNA

So far: all results are assuming UNA (the Unique Name Assumption)
⇒ also assumed by the underlying relational technology.

BUT OWL does NOT adopt UNA...

What happens without UNA?
UNA or not UNA

So far: all results are assuming UNA (the Unique Name Assumption) ⇒ also assumed by the underlying relational technology.

BUT OWL does NOT adopt UNA . . .

What happens without UNA?

$\text{DL-Lite}_{\text{core}}^V$ data complexity: $\text{coNP}$ (Artale et al. 2009)

$\text{DL-Lite}_{\text{horn}}$ data complexity: PTIME-complete

⇒ explicit account of equality (via an auxiliary relation $eq$) added to the construction of $\mathcal{I}_V$ (doesn’t affect queries)
UNA or not UNA

So far: all results are assuming UNA (the Unique Name Assumption) ⇒ also assumed by the underlying relational technology.

BUT OWL _does NOT adopt UNA_ . . .

**What happens without UNA?**

| DL-Lite$^\mathcal{N}_{\text{core}}$ | data complexity: coNP (Artale et al. 2009) |
| DL-Lite$^\mathcal{E}_{\text{horn}}$ | data complexity: PTIME-complete |

⇒ explicit account of equality (via an auxiliary relation eq)  
⇒ added to the construction of $\mathcal{I}_\mathcal{K}$ (doesn’t affect queries)
Experiments

Ontologies:

- Galen-lite (2733 concepts, 207 roles, 4888 axioms)
- Core (81 concept names, 58 roles, and 381 axioms)
- Stockexchange (17 concepts, 12 roles, 62 axioms)
- University (31 concepts, 25 roles, 103 axioms)

System:

DB2-Express version 9.5 running on Intel Core 2 Duo 2.5GHz CPU, 4GB memory and 500GB storage under Linux 2.6.28.

Queries:

Conjunctive queries with 3-6 atoms in their bodies, e.g.,

Q1(x) :- horn(x), hasstate(x,y), cellmorphologystate(y).
Q2(x) :- shortbone(x), hasstate(x,y), cellmorphologystate(y).
Q3(x) :- tissue(x), hasstate(x,y), temporalunit(y).
Q4(x) :- protozoa(x), contains(x,y), metal(y),
contains(x,z), steroid(z).
## Experiments (results)

<table>
<thead>
<tr>
<th>ABox size (in M)</th>
<th>query</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>canonical</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>CA</td>
<td>RA</td>
<td>CA</td>
<td>RA</td>
<td>UN</td>
</tr>
<tr>
<td><strong>Galen-Lite</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2.0</td>
<td>2.0</td>
<td>9.9</td>
<td>3.7</td>
</tr>
<tr>
<td>50</td>
<td>5.0</td>
<td>5.0</td>
<td>24.8</td>
<td>9.3</td>
</tr>
<tr>
<td>70</td>
<td>10.0</td>
<td>10.0</td>
<td>43.0</td>
<td>15.4</td>
</tr>
<tr>
<td>100</td>
<td>20.0</td>
<td>20.0</td>
<td>75.0</td>
<td>25.8</td>
</tr>
</tbody>
</table>

Legend:
- CA-number of concept assertions; RA-number of role assertions
- UN-original query; RW-canonical interpretation; QO-QuOnto system
Summary of Contributions

Contributions

1. Combined approach to query answering in DL-Lite
   ⇒ efficiency gains in comparison with pure rewriting,
   ⇒ non-UNA in DL-Lite$^F$ can be supported.

2. Polynomial rewriting for DL-Lite$^F_{\text{core}}$.

Future Work

1. Better integration with role hierarchies
   ⇒ we can do this efficiently (but not by poly-sized query)

2. Incremental update of the canonical interpretation
   ⇒ using techniques for incremental view maintenance
Summary of Contributions

Contributions

1. Combined approach to query answering in DL-Lite
   \[\Rightarrow\] efficiency gains in comparison with pure rewriting,
   \[\Rightarrow\] non-UNA in DL-Lite\(^F\) can be supported.

2. Polynomial rewriting for DL-Lite\(^F\)core.

Future Work

1. Better integration with role hierarchies
   \[\Rightarrow\] we can do this efficiently (but not by poly-sized query)

2. Incremental update of the canonical interpretation
   \[\Rightarrow\] using techniques for incremental view maintenance