The Combined Approach to Query Answering in DL-Lite

Roman Kontchakov[1], Carsten Lutz[2], David Toman[3], Frank Wolter[4], and Michael Zakharyaschev[1]

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

[2]Fachbereich Mathematik und Informatik, Universität Bremen, Germany
clu@informatik.uni-bremen.de

[3]D.R. Cheriton School of Computer Science, University of Waterloo, Canada
david@cs.uwaterloo.ca

[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)

$\Rightarrow$ *large data sets* and *(relatively) large ontologies*.

$\Rightarrow$ 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;

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)

$\Rightarrow$ *large data sets* and *(relatively) large ontologies*.

$\Rightarrow$ 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)

$\Rightarrow$ *large data sets* and *(relatively) large ontologies*.

$\Rightarrow$ 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: $\underbrace{\text{Ontology } (\mathcal{T}),\ \text{Data } (\mathcal{A})}_{\text{Knowledge Base}(\mathcal{K})}$, and a Query ($Q$)

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

Approaches:

1. Reduction to *standard reasoning* (e.g., satisfiability)
2. Reduction to *querying a relational database*

   $\Rightarrow$ *very* good at $\{a \mid \mathcal{A} \models Q[a]\}$ for range restricted Q

   . . . what do we do with $\mathcal{T}$?

Approaches to Ontology-based Data Access

Main Task

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

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

Approaches:

1. Reduction to *standard reasoning* (e.g., satisfiability)
2. Reduction to *querying a relational database*

   $\Rightarrow$ *very* good at $\{a \mid \mathcal{A} \models Q[a]\}$ for range restricted Q

   ...what do we do with $\mathcal{T}$?

Definitions&Background

Definition (DL-Lite$^{\mathcal{N}}_{\text{horn}}$)

$$\text{roles: } R ::= P \mid P^-, \qquad \text{concepts: } C ::= \bot \mid A \mid \geq m\, R.$$

where $P \in \mathsf{N_R}$, $A \in \mathsf{N_C}$ and $m > 0$.

1. An *ontology (TBox)* is a finite set $\mathcal{T}$ of *concept* inclusions $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq C$;
2. The *Data (ABox)* is a finite set $\mathcal{A}$ 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*.
   $\Rightarrow$ make *implicit knowledge explicit (data completion)*.
2. Use the *data completion* (only) to answer queries
   $\Rightarrow$ 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)
   $\Rightarrow$ we define a *canonical interpretation* $\mathcal{I}_{\mathcal{K}}$ (representatives)
2. Can we then use the original Conjunctive Query?
   Not directly: $Q(\mathcal{I}_{\mathcal{K}})$ can produce *"spurious matches"*
   $\Rightarrow$ 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., $\mathcal{T}$) into the *data*.

   $\Rightarrow$ make *implicit knowledge* *explicit (data completion)*.

2. Use the *data completion* (only) to answer queries

   $\Rightarrow$ and use a relational system to do this *efficiently*.

Example

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

1. $\mathcal{I}_{\mathcal{K}} = \{BOSS(Bob), EMP(Bob)\}$ (data completion)
2. $Q(\mathcal{I}_{\mathcal{K}}) = \{Bob\}$ (relational query)

The Master Plan

IDEA:

1. Incorporate the background knowledge (i.e., $\mathcal{T}$) into the *data*.
   $\Rightarrow$ make *implicit knowledge explicit (data completion)*.
2. Use the *data completion* (only) to answer queries
   $\Rightarrow$ 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)
   $\Rightarrow$ we define a *canonical interpretation* $\mathcal{I}_{\mathcal{K}}$ (representatives)
2. Can we then use the original Conjunctive Query?
   Not directly: $Q(\mathcal{I}_{\mathcal{K}})$ can produce *"spurious matches"*
   $\Rightarrow$ we eliminate the spurious matches by rewriting the query (independently of $\mathcal{T}$ and $\mathcal{A}$)

Canonical Interpretations

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

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

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

... c_R's only used "when necessary" (for *generating* roles)

Lemma

There are queries

- $q_A^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_A^\mathcal{T}, \mathcal{A}) = A^{\mathcal{I}_\mathcal{K}}$, *and*
- $q_P^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_P^\mathcal{T}, \mathcal{A}) = P^{\mathcal{I}_\mathcal{K}}$

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

free consistency test: $q_\bot^\mathcal{T}(\mathcal{A}) = \emptyset$

Canonical Interpretations

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

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

$$P^{\mathcal{I}_\mathcal{K}} = \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid P(a,b) \in \mathcal{A}\} \cup \{(d, c_P) \in \Delta^{\mathcal{I}_\mathcal{K}} \times \mathsf{N}_\mathsf{I}^\mathcal{T} \mid d \leadsto c_P\} \cup \{(c_{P^-}, d) \in \mathsf{N}_\mathsf{I}^\mathcal{T} \times \Delta^{\mathcal{I}_\mathcal{K}} \mid d \leadsto c_{P^-}\}$$

... c_R's only used "when necessary" (for *generating* roles)

Lemma

There are queries

- $q_A^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_A^\mathcal{T}, \mathcal{A}) = A^{\mathcal{I}_\mathcal{K}}$, *and*
- $q_P^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_P^\mathcal{T}, \mathcal{A}) = P^{\mathcal{I}_\mathcal{K}}$

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

free consistency test: $q_\bot^\mathcal{T}(\mathcal{A}) = \emptyset$

Canonical Interpretations

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

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

$$P^{\mathcal{I}_\mathcal{K}} = \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid P(a,b) \in \mathcal{A}\} \cup \{(d, c_P) \in \Delta^{\mathcal{I}_\mathcal{K}} \times \mathsf{N}_\mathsf{I}^\mathcal{T} \mid d \leadsto c_P\} \cup \{(c_{P^-}, d) \in \mathsf{N}_\mathsf{I}^\mathcal{T} \times \Delta^{\mathcal{I}_\mathcal{K}} \mid d \leadsto c_{P^-}\}$$

... c_R's only used "when necessary" (for *generating* roles)

Lemma

There are queries

- $q_A^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_A^\mathcal{T}, \mathcal{A}) = A^{\mathcal{I}_\mathcal{K}}$, *and*
- $q_P^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_P^\mathcal{T}, \mathcal{A}) = P^{\mathcal{I}_\mathcal{K}}$

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

Canonical Interpretations

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

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

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

. . . c_R's only used "when necessary" (for *generating* roles)

Example

$$\mathcal{T} = \{EMP \sqsubseteq \exists MANAGES, \exists MANAGES^- \sqsubseteq BOSS, BOSS \sqsubseteq EMP\}$$
$$\mathcal{A} = \{EMP(Bob), EMP(Sue)\}$$

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

$\mathcal{I}_\mathcal{K}$ is NOT model of $(\mathcal{T}, \mathcal{A})$ in general.

Canonical Interpretations

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

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

$$P^{\mathcal{I}_\mathcal{K}} = \{(a,b) \in \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mid P(a,b) \in \mathcal{A}\} \cup \{(d, c_P) \in \Delta^{\mathcal{I}_\mathcal{K}} \times \mathsf{N}_\mathsf{I}^\mathcal{T} \mid d \leadsto c_P\} \cup \{(c_{P^-}, d) \in \mathsf{N}_\mathsf{I}^\mathcal{T} \times \Delta^{\mathcal{I}_\mathcal{K}} \mid d \leadsto c_{P^-}\}$$

... c_R's only used "when necessary" (for *generating* roles)

Example

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

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

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

$\mathcal{I}_\mathcal{K}$ is NOT model of $(\mathcal{T}, \mathcal{A})$ in general.

Canonical Interpretations

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

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

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

. . . c_R's only used "when necessary" (for *generating* roles)

Lemma

There are queries

- $q_A^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_A^\mathcal{T}, \mathcal{A}) = A^{\mathcal{I}_\mathcal{K}}$, *and*
- $q_P^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_P^\mathcal{T}, \mathcal{A}) = P^{\mathcal{I}_\mathcal{K}}$

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

free consistency test: $q_\bot^\mathcal{T}(\mathcal{A}) = \emptyset$

Canonical Interpretations

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

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

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

... c_R's only used "when necessary" (for *generating* roles)

Lemma

There are queries

- $q_A^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_A^\mathcal{T}, \mathcal{A}) = A^{\mathcal{I}_\mathcal{K}}$, *and*
- $q_P^\mathcal{T}$ *s.t.* $\mathsf{ans}(q_P^\mathcal{T}, \mathcal{A}) = P^{\mathcal{I}_\mathcal{K}}$

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

free consistency test: $q_\bot^\mathcal{T}(\mathcal{A}) = \emptyset$

Query Rewriting

Example

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

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

Queries:

1. $\exists v.MANAGES(v, v)$
2. $\exists y.MANAGES(x, y) \wedge MANAGES(z, y)$

Query Rewriting

$$\exists \bar{u}.\varphi \quad \mapsto \quad \exists \bar{u}.\varphi \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3$$

where
- φ_1 eliminates answers containing c_R's;
- φ_2 eliminates problem (1) above; and
- φ_3 eliminates problem (2) above.

} *selections* in SQL

Query Rewriting

Example

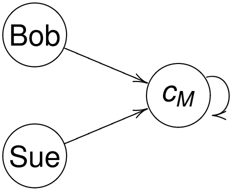
$\mathcal{T} = \{EMP \sqsubseteq \exists MANAGES, \exists MANAGES^{-} \sqsubseteq BOSS, BOSS \sqsubseteq EMP\}$

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

Queries:

1. $\exists v.MANAGES(v, v)$
2. $\exists y.MANAGES(x, y) \wedge MANAGES(z, y)$

$\mathcal{I}_{\mathcal{K}}$:



$Q_1(\mathcal{I}_{\mathcal{K}}) = \{c_M\}$
$Q_2(\mathcal{I}_{\mathcal{K}}) = \{(Bob, Sue)\}$

Query Rewriting

Example

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

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

Queries:

1. $\exists v.MANAGES(v, v)$
2. $\exists y.MANAGES(x, y) \land MANAGES(z, y)$

Query Rewriting

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

where
- φ_1 eliminates answers containing c_R's;
- φ_2 eliminates problem (1) above; and
- φ_3 eliminates problem (2) above.

} *selections* in SQL

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)

$\Rightarrow$ 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)

data complexity: PTIME-complete

$\Rightarrow$ explicit account of equality (via an auxiliary relation eq)

$\Rightarrow$ added to the construction of $\mathcal{T}_P$ (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?

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

$\text{DL-Lite}_{\text{horn}}^{\mathcal{F}}$ 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)

	Ind (in K)	ABox size (in M)				query															
		original		canonical		*Q1*			*Q2*			*Q3*			*Q4*						
		CA	RA	CA	RA	UN	RW	QO	UN	RW	QO	UN	RW	QO	UN	RW	QO				
Galen-Lite	20	2.0	2.0	9.9	3.7	0.02	0.04	13.69	0.02	0.08	1.65	0.02	0.11	1m 28	0.12	0.22	16m 11				
	50	5.0	5.0	24.8	9.3	0.04	0.55	14.39	0.05	0.19	2.21	0.03	0.28	51.39	0.11	0.43	13m 26				
	70	10.0	10.0	43.0	15.4	0.03	0.76	17.56	0.11	0.55	3.01	0.06	0.73	1m 11	0.15	0.63	13m 00				
	100	20.0	20.0	75.0	25.8	0.05	0.87	23.86	0.14	0.76	6.55	0.12	0.95	1m 31	0.18	1.52	16m 23				

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$^{\mathcal{F}}$ can be supported.
2. Polynomial rewriting for DL-Lite$^{\mathcal{F}}_{\text{core}}$.

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
   - ⇒ efficiency gains in comparison with pure rewriting,
   - ⇒ non-UNA in DL-Lite$^{\mathcal{F}}$ can be supported.
2. Polynomial rewriting for DL-Lite$^{\mathcal{F}}_{\text{core}}$.

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*