Description Logics

Franz Baader
Theoretical Computer Science
TU Dresden
Germany

1. Motivation and introduction to Description Logics
2. Tableau-based reasoning procedures
3. Automata-based reasoning procedures
4. Complexity of reasoning in Description Logics
5. Reasoning in inexpressive Description Logics
6. Query answering in inexpressive Description Logics

Dresden

Query Answering

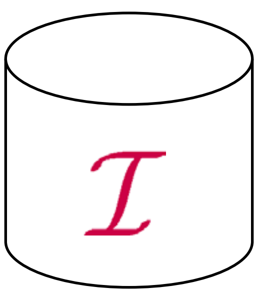
in databases from a logical point of view

The problem:

- A database is a finite first-order interpretation (i.e., a finite relational structure).
- A query is a first-order formula with some free variables (the answer variables).
- An answer tuple assigns elements of the interpretation to the free variables such that the query is satisfied.

FOL query

$$\phi(x_1, \ldots, x_n)$$



Answer tuples:

$$\{(d_1, \ldots, d_n) \mid \mathcal{I} \models \phi(d_1, \ldots, d_n)\}$$

Dresden

Query Answering

in databases from a logical point of view

The problem:

- A database is a finite first-order interpretation (i.e., a finite relational structure).
- A query is a first-order formula with some free variables (the answer variables).

FOL query

- An answer tuple assigns elements of the interpretation to the free variables such that the query is satisfied.

Its complexity: deciding whether there is an answer tuple is

- PSpace-complete w.r.t. combined complexity, i.e., w.r.t. the combined size of $\mathcal{I}$ and ϕ.
- In AC^0 w.r.t. data complexity, i.e., w.r.t. the size of $\mathcal{I}$ only (ϕ fixed).

Usually $\mathcal{I}$ is very large and ϕ quite small.

$$AC^0 \subset \text{LogSpace} \subseteq \text{P}$$

Dresden

Query Answering

in databases from a logical point of view

The problem:

- A database is a finite first-order interpretation
  (i.e., a finite relational structure).
- A query is a first-order formula with some free variables
  (the answer variables).

FOL query

- An answer tuple assigns elements of the interpretation
  to the free variables such that the query is satisfied.

In practice:

- highly efficient relational database engines available
- that scale very well to huge databases

Conjunctive queries

subclass of FOL queries

A conjunctive query (CQ)

is an existentially quantified conjunction of atoms:

$$\exists y, z.\ \ Q(x) \wedge P(x, y) \wedge P(y, z) \wedge P(z, x)$$

A union of conjunctive queries (UCQ) is a disjunction of CQs.

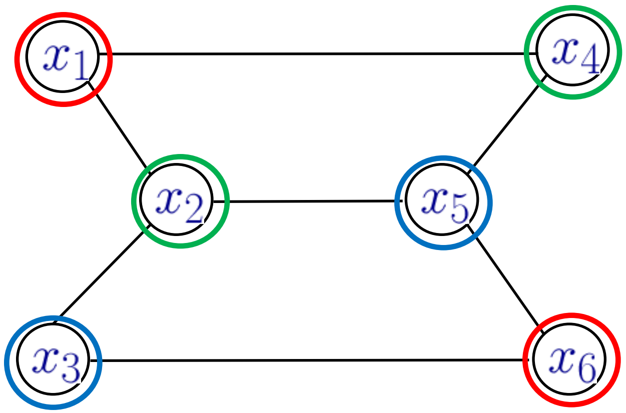
Complexity of CQs and UCQs:
deciding whether there is an answer tuple is

- NP-complete w.r.t. combined complexity.
- In AC^0 w.r.t. data complexity.

Dresden

Conjunctive queries

example that shows NP-hardness w.r.t. combined complexity



three-colorability

Conjunctive query:

$$\exists x_1, x_2, x_3, x_4, x_5, x_6.$$
$$E(x_1, x_2) \wedge E(x_2, x_3) \wedge$$
$$E(x_1, x_4) \wedge E(x_2, x_5) \wedge E(x_3, x_6) \wedge$$
$$E(x_4, x_5) \wedge E(x_5, x_6)$$

Database:

$E(\textit{red}, \textit{blue})$

$E(\textit{red}, \textit{green})$

$E(\textit{green}, \textit{red})$

The empty tuple () is an answer tuple iff the graph is three-colorable.

Dresden

Ontology-Based Data Access

OBDA

Generalizes answering (unions of) conjunctive queries in two directions:

- Presence of a TBox $\mathcal{T}$:
  predicates used in the CQs are constrained by TBox axioms
- Incompleteness:
  CQs evaluated over an ABox $\mathcal{A}$ rather than an interpretation
  (no closed-word assumption).

We want to compute certain answers of $\phi(x_1, \ldots, x_n)$ over $\mathcal{A}$ w.r.t. $\mathcal{T}$:

- a tuple $(a_1, \ldots, a_n)$ of individuals occurring in $\mathcal{A}$ such that
- $(a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}})$ is an answer tuple of $\phi(x_1, \ldots, x_n)$ over $\mathcal{I}$
- for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$.

Ontology-Based Data Access

- One usually assumes that the ABox $\mathcal{A}$ is atomic, i.e., contains only atomic assertions of the form $A(a), r(a, b)$ for concept names A and role names r.

- Data complexity: complexity of computing certain answers in the size of the ABox only (TBox and query assumed to be fixed).

- Combined complexity: complexity of computing certain answers in the size of the ABox, TBox, and query.

- The instance problem can be seen as a special case:
  $\mathcal{A} \models_{\mathcal{T}} A(e)$ iff (e) is a certain answer of $A(x)$ over $\mathcal{A}$ w.r.t. $\mathcal{T}$.

  ⇨ Combined complexity of computing certain answers at least as high as the complexity of the instance problem.

Dresden

Ontology-Based Data Access for expressive DLs

For the DL $\mathcal{ALC}$, deciding whether there is a certain answer is

- ExpTime-complete w.r.t. combined complexity. *same as instance problem*
- coNP-complete w.r.t. data complexity. *not even tractable*

Adding inverse roles increases the combined complexity:
for the DL $\mathcal{ALCI}$, deciding whether there is a certain answer is

- 2ExpTime-complete w.r.t. combined complexity. *higher than instance problem*
- coNP-complete w.r.t. data complexity.

Ontology-Based Data Access

for inexpressive DLs

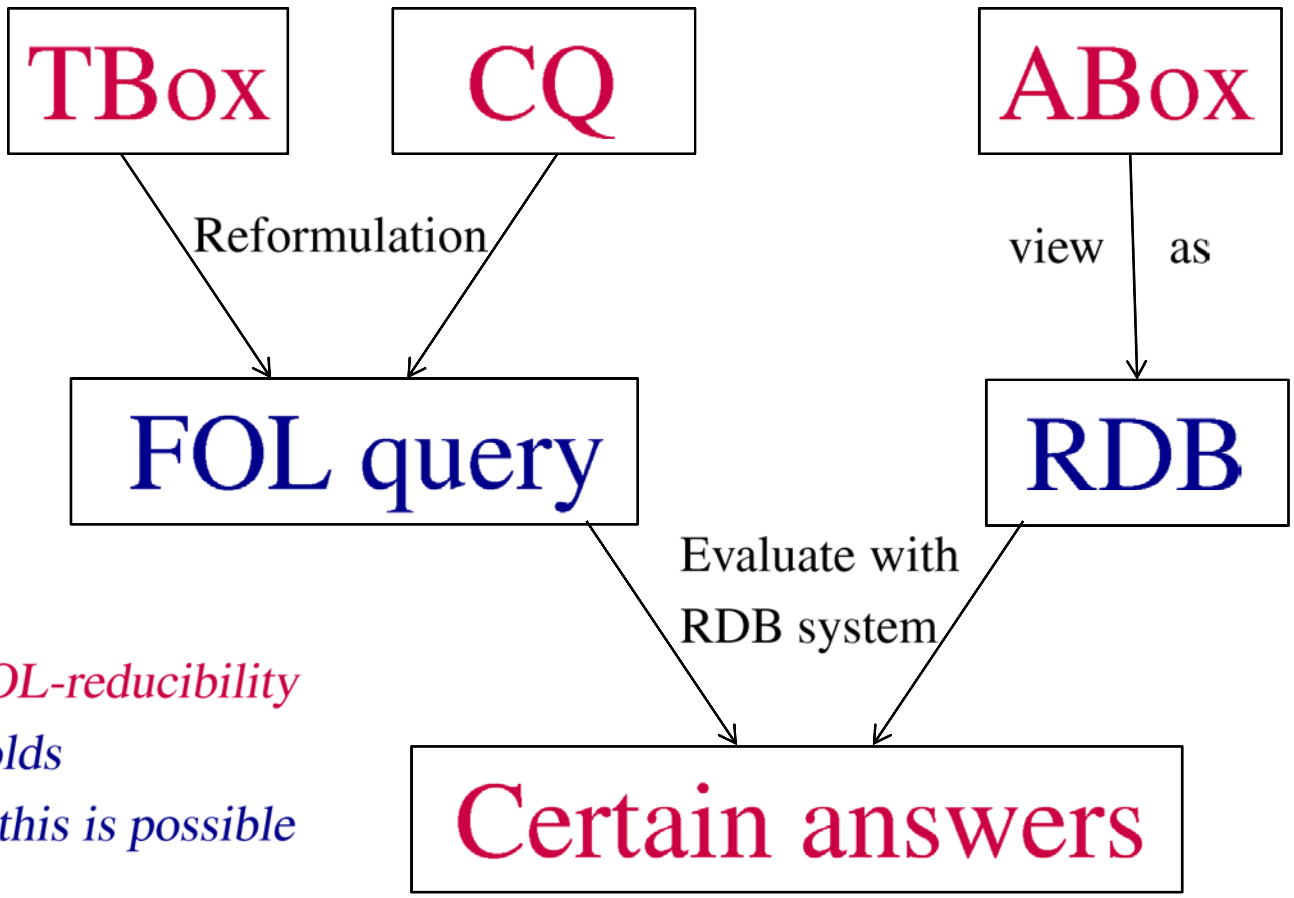
In order to deal with very large ABoxes, tractability is not sufficient.

Goal

Find DLs for which computing certain answers can be reduced to answering FOL queries using a relational database system.

Query answering

using relational DB technology



FOL-reducibility
holds
if this is possible

Ontology-Based Data Access for inexpressive DLs

In order to deal with very large ABoxes,
tractability is not sufficient.

Goal

Find DLs for which computing certain answers
caFind DLs for which FOL-reducibility holds.
using a relational database system.

⇒ the DL-Lite family

Dresden

DL-Lite$_{core}$

the basic member of the DL-Lite family [Calvanese et al.; 2007]

concept names	A
basic concepts	B
general concepts	C

$$B \;\rightarrow\; A \mid \exists r.\top \mid \exists r^{-1}.\top$$
$$C \;\rightarrow\; B \mid \neg B$$

GCIs

$$B \sqsubseteq C$$

$$\exists has\_child.\top \sqsubseteq \neg Spinster$$
$$\exists has\_child.\top \sqsubseteq Parent$$
$$Parent \sqsubseteq Human$$
$$Human \sqsubseteq \exists has\_child^{-1}.\top$$

ABox

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

$$Woman(LINDA)$$
$$has\_child(LINDA, JAMES)$$
$$Beatle(PAUL)$$
$$has\_child(PAUL, JAMES)$$

Conjunctive query answering

over a DL-Lite$_{core}$ ontology

$$\exists y, z_1, z_2. Woman(x) \wedge has\_child(x, y) \wedge has\_child(z_1, y) \wedge Human(z_1) \wedge has\_child(z_2, z_1)$$

free variable

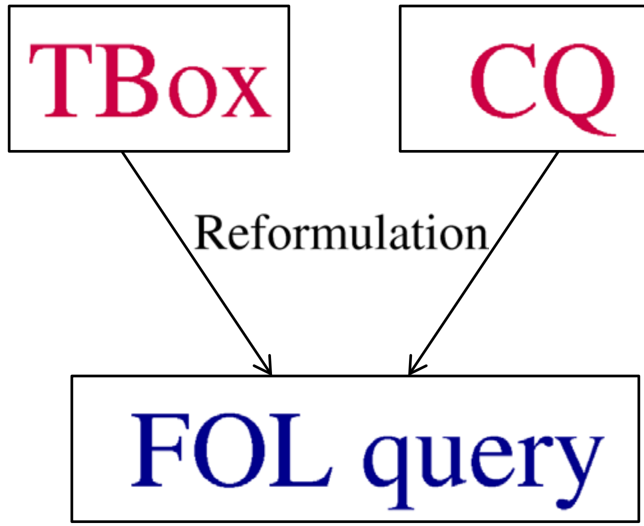
certain answer: $(LINDA)$

TBox		ABox
$\exists has\_child.\top \sqsubseteq \neg Spinster$	ontology	$Woman(LINDA)$
$\exists has\_child.\top \sqsubseteq Parent$		$has\_child(LINDA, JAMES)$
$Parent \sqsubseteq Human$		$Beatle(PAUL)$
$Human \sqsubseteq \exists has\_child^{-1}.\top$		$has\_child(PAUL, JAMES)$

Dresden

FOL-reducibility

of DL-Lite$_{core}$ [Calvanese et al.; 2007]



Query reformulation generates a union of conjunctive queries by

- using GCIs with basic concepts on right-hand side as rewrite rules from right to left,
- which generate a new CQ in the union by rewriting an atom in an already obtained CQ.

FOL-reducibility of DL-Lite$_{core}$

$$\exists y, z_1, z_2. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Human(z_1) \wedge has\_child(z_2,z_1)$$

$$\exists y, z_1. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Human(z_1) \wedge Human(z_1)$$

TBox

$\exists has\_child.\top \sqsubseteq \neg Spinster$ $\exists has\_child.\top \sqsubseteq Parent$

$Parent \sqsubseteq Human$ $Human \sqsubseteq \exists has\_child^{-1}.\top$

Dresden

FOL-reducibility of DL-Lite$_{core}$

$$\exists y, z_1, z_2. Woman(x) \land has\_child(x,y) \land has\_child(z_1,y) \land Human(z_1) \land has\_child(z_2,z_1)$$

$$\exists y, z_1. Woman(x) \land has\_child(x,y) \land has\_child(z_1,y) \land \boxed{Human(z_1)}$$

$$\exists y, z_1. Woman(x) \land has\_child(x,y) \land has\_child(z_1,y) \land Parent(z_1)$$

TBox

$\exists has\_child.\top \sqsubseteq \neg Spinster$ $\quad$ $\exists has\_child.\top \sqsubseteq Parent$

$\boxed{Parent \sqsubseteq Human}$ $\quad$ $Human \sqsubseteq \exists has\_child^{-1}.\top$

Dresden

FOL-reducibility of DL-Lite$_{core}$

$$\exists y, z_1, z_2. Woman(x) \wedge has\_child(x, y) \wedge has\_child(z_1, y) \wedge Human(z_1) \wedge has\_child(z_2, z_1)$$

$$\exists y, z_1. Woman(x) \wedge has\_child(x, y) \wedge has\_child(z_1, y) \wedge Human(z_1)$$

$$\exists y, z_1. Woman(x) \wedge has\_child(x, y) \wedge has\_child(z_1, y) \wedge \boxed{Parent(z_1)}$$

$$\exists y, z_1, z_3. Woman(x) \wedge has\_child(x, y) \wedge has\_child(z_1, y) \wedge has\_child(z_1, z_3)$$

TBox

$\exists has\_child.\top \sqsubseteq \neg Spinster$ $\quad$ $\boxed{\exists has\_child.\top \sqsubseteq Parent}$

$Parent \sqsubseteq Human$ $\quad$ $Human \sqsubseteq \exists has\_child^{-1}.\top$

Dresden

FOL-reducibility of DL-Lite$_{core}$

$$\exists y, z_1, z_2. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Human(z_1) \wedge has\_child(z_2,z_1)$$

$$\exists y, z_1. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Human(z_1)$$

$$\exists y, z_1. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Parent(z_1)$$

$$\exists y, z_1, z_3. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge has\_child(z_1,z_3)$$

ABox

$Woman(LINDA)$ $has\_child(LINDA, JAMES)$
$Beatle(PAUL)$ $has\_child(PAUL, JAMES)$

TBox

$\exists has\_child.\top \sqsubseteq \neg Spinster$ $\exists has\_child.\top \sqsubseteq Parent$
$Parent \sqsubseteq Human$ $Human \sqsubseteq \exists has\_child^{-1}.\top$

FOL-reducibility of DL-Lite$_{core}$

$$\exists y, z_1, z_2. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Human(z_1) \wedge has\_child(z_2,z_1)$$

$$\exists y, z_1. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Human(z_1)$$

$$\exists y, z_1. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge Parent(z_1)$$

$$\exists y, z_1, z_3. Woman(x) \wedge has\_child(x,y) \wedge has\_child(z_1,y) \wedge has\_child(z_1,z_3)$$

RDB

$Woman(LINDA)$ $has\_child(LINDA, JAMES)$
$Beatle(PAUL)$ $has\_child(PAUL, JAMES)$

answer tuple: $(LINDA)$

FOL-reducibility of DL-Lite$_{core}$

Some subtleties

- When rewriting with existential restrictions, the variable that "is lost" should not occur anywhere else.

$$\exists y, z_1, z_2. Woman(x) \wedge has\_child(x, y) \wedge has\_child(z_1, y) \wedge Human(z_1) \wedge \boxed{has\_child(z_2, z_1)}$$

$$Human \sqsubseteq \exists has\_child^{-1}.\top$$

- To satisfy this constraint, one sometimes needs to unify atoms.

$$\exists y, z_1. has\_child(x, y) \wedge has\_child(z_1, y)$$

$$Parent \sqsubseteq \exists has\_child.\top$$

Unification replaces z_1 by x: $\exists y. has\_child(x, y)$

$$Parent(x)$$

Dresden

FOL-reducibility

for the DL-Lite family of DLs

- DL-Lite$_{core}$ and its extensions DL-Lite$_{\mathcal{R}}$ and DL-Lite$_{\mathcal{F}}$ are FOL-reducible.

additional role inclusion axioms:

$r_1 \sqsubseteq r_2$

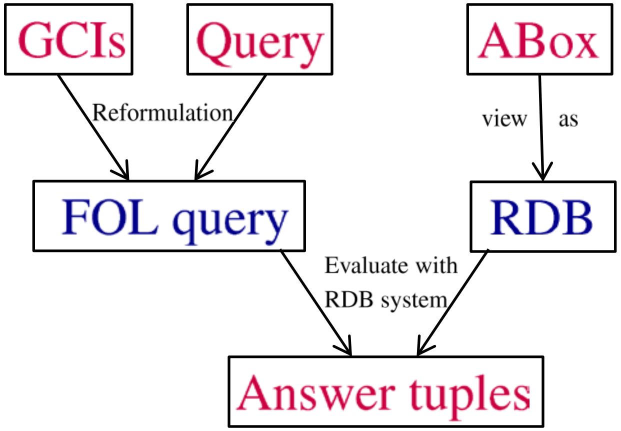
$r_1 \sqsubseteq \neg r_2$

additional functionality axioms:

$\top \sqsubseteq (\leq 1\, r)$

$\top \sqsubseteq (\leq 1\, r^{-1})$

- FOL-reducibility implies a data complexity in AC^0 for query answering, and thus in particular tractability w.r.t. data complexity.



Dresden

FOL-reducibility

for the DL-Lite family of DLs

- DL-Lite$_{core}$ and its extensions DL-Lite$_{\mathcal{R}}$ and DL-Lite$_{\mathcal{F}}$ are FOL-reducible.

additional role inclusion axioms:

$r_1 \sqsubseteq r_2$

$r_1 \sqsubseteq \neg r_2$

additional functionality axioms:

$\top \sqsubseteq (\leq 1\, r)$

$\top \sqsubseteq (\leq 1\, r^{-1})$

- FOL-reducibility implies a data complexity in AC^0 for query answering, and thus in particular tractability w.r.t. data complexity.

- DL-Lite$_{\mathcal{R}}$ is the formal basis for the OWL 2 QL profile of the new OWL 2 standard

- Approach implemented in the QUONTO system.

Dresden

Query answering in $\mathcal{EL}$

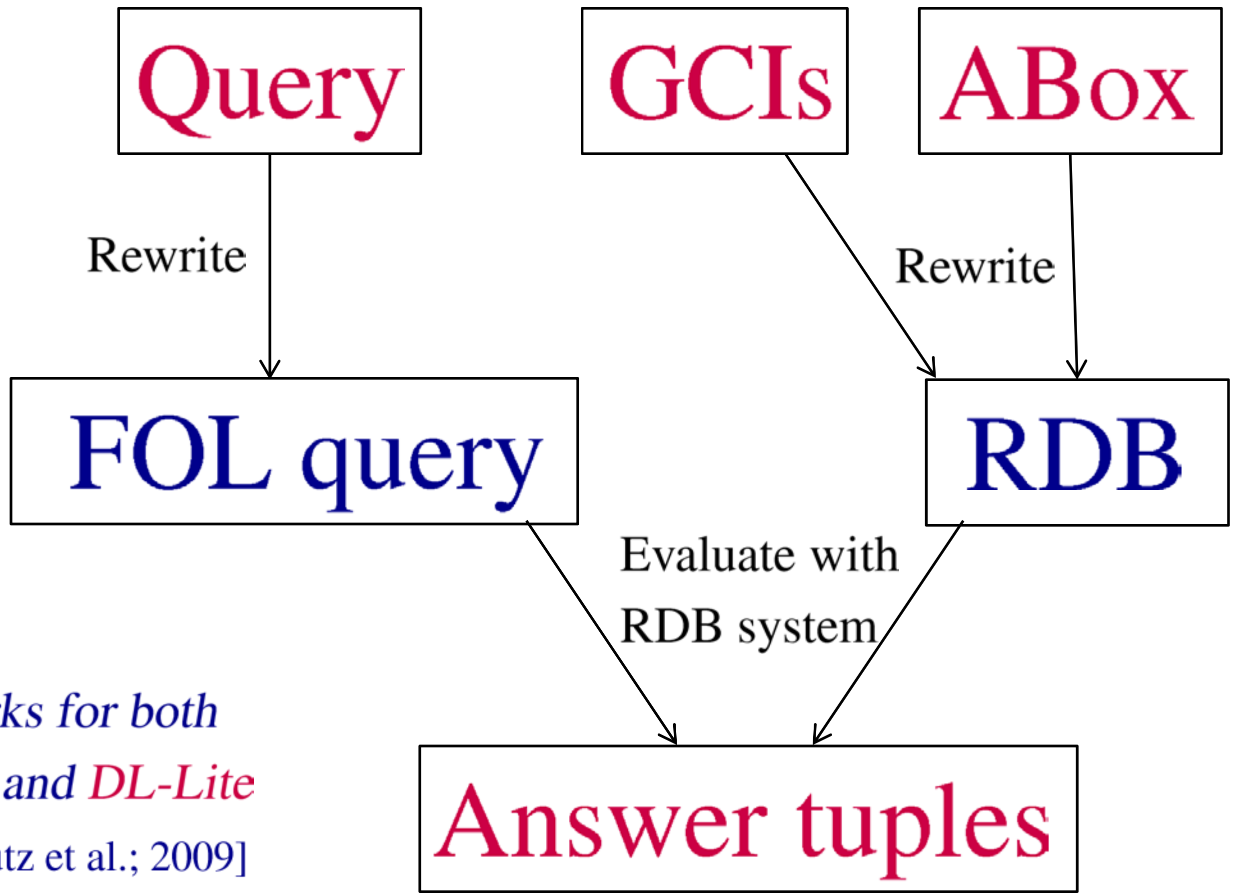
- Computing certain answers w.r.t. $\mathcal{EL}$-TBoxes is tractable w.r.t. data complexity.
- More precisely, it is PTime-complete, and thus not in AC^0.
- Thus, query answering in $\mathcal{EL}$ is not FOL-reducible.

Can we still use RDB technology for query evaluation?

Query answering

in $\mathcal{EL}$ using relational DB technology

beyond FOL-reducibility

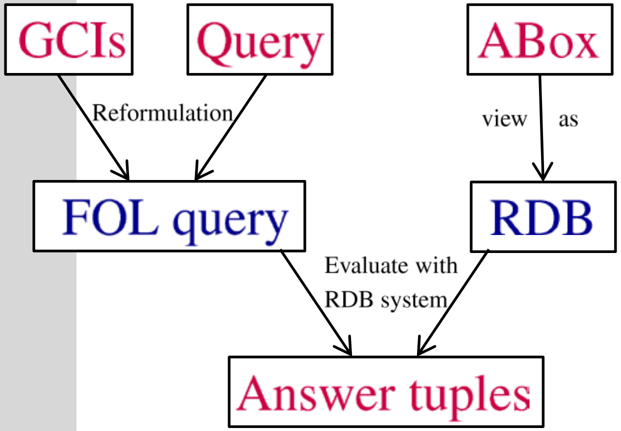


works for both

$\mathcal{EL}$ and DL-Lite

[Lutz et al.; 2009]

[Calvanese et al.; 2007]

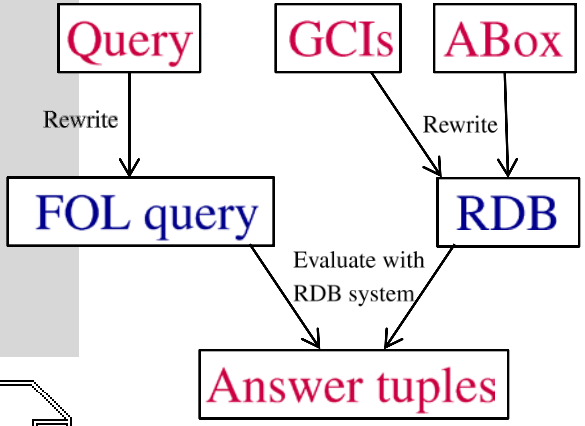


+ Query reformulation independent of ABox.

+ ABox need not be changed.

– Size of reformulated query may grow exponentially.

[Lutz et al.; 2009]



+ Query rewriting independent of ABox and GCIs.

– ABox needs to be changed.

+ ABox rewriting independent of query.

+ Both ABox and query rewriting polynomial.

Description Logics

Franz Baader

Theoretical Computer Science

TU Dresden

Germany

Literature:

- F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (ed.): The Description Logic Handbook. Cambridge University Press, 2003.

- F. Baader: Description Logics. In Reasoning Web 2009, Springer LNCS 5689, 2009.

- F. Baader, C. Lutz, and A.-Y. Turhan: Small is again beautiful in Description Logics. KI — Künstliche Intelligenz, 24(1):2533, 2010.

Dresden