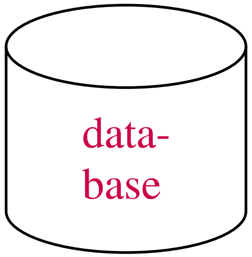
Chapter 7

Query answering

We first consider query answering in databases from a logical point of view, and then extend this to ontology-mediated query answering in order to

- allow for incomplete data;
- take background knowledge into account;
- deal with potentially infinite data sets.



finite collection of relations over a finite domain

SQL query describing answer tuples

DB = finite interpretation

basically same expressiveness

interpretation $\mathcal{I}$ over a relational signature

domain $\Delta^{\mathcal{I}}$ together with relations interpreting the relation symbols

First-order formula $\phi(x_1, \ldots, x_n)$ with free variables (answer variables)

Dresden

First-order queries

We restrict the attention to queries with unary and binary relation symbols corresponding to concepts and roles in DLs.

We give the definitions for arbitrary interpretations, not just finite ones.

Definition 7.1 (FO query)

An FO query is a first-order formula that uses only unary and binary predicates (concept and role names), and no function symbols or constants. The use of equality is allowed.

The free variables $\vec{x}$ of an FO query $q(\vec{x})$ are called answer variables.

The arity of $q(\vec{x})$ is the number of answer variables.

Let $q(\vec{x})$ be an FO query of arity k and $\mathcal{I}$ an interpretation. We say that

$\vec{a} = a_1, \ldots, a_k$ is an answer to q on $\mathcal{I}$ if $\mathcal{I} \models q[\vec{a}]$

i.e., if $q(\vec{x})$ evaluates to true in $\mathcal{I}$ under the valuation that interprets the answer variables $\vec{x}$ as the constants $\vec{a}$.

$\mathsf{ans}(q, \mathcal{I})$: set of all answers to q in $\mathcal{I}$

Dresden

Conjunctive queries

restricted class of FO queries

Definition 7.2 (conjunctive query)

A conjunctive query (CQ) q has the form

$$\exists x_1 \cdots \exists x_k \, (\alpha_1 \wedge \cdots \wedge \alpha_n)$$

where $k \geq 0$, $n \geq 1$, $x_1, \ldots, x_k$ are variables, and each α_i is a concept atom $A(x)$ or a role atom $r(x, y)$ with $A \in \mathbf{C}$, $r \in \mathbf{R}$, and x, y variables.

We call $x_1, \ldots, x_k$ the quantified variables and all other variables in q the answer variables.

The arity of q is the number of answer variables.

To express that the answer variables in a CQ q are $\vec{x}$, we often write $q(\vec{x})$ instead of just q.

Dresden

Conjunctive queries

examples

where answer variables are underlined

$$q_1(x_1, x_2) = \mathsf{Professor}(\underline{x_1}) \land \mathsf{supervises}(\underline{x_1}, \underline{x_2}) \land \mathsf{Student}(\underline{x_2})$$

Returns all pairs of constants (i.e., individual names) (a, b) such that a is a professor who supervises the student b.

$$q_2(x) = \exists y\ (\mathsf{Professor}(y) \land \mathsf{supervises}(y, \underline{x}) \land \mathsf{Student}(\underline{x}))$$

Returns all individual names a such that a is a student supervised by some professor.

$$q_3(x_1, x_2) = \exists y\ (\mathsf{Professor}(y) \land \mathsf{supervises}(y, \underline{x_1}) \land \mathsf{supervises}(y, \underline{x_2}) \land \mathsf{Student}(\underline{x_1}) \land \mathsf{Student}(\underline{x_2}))$$

Returns all pairs of students supervised by the same professor.

Conjunctive queries

characterisation of answer tuples

Definition 7.3 ($\vec{a}$-match)

Let q be a conjunctive query and $\mathcal{I}$ an interpretation.
We use $\mathsf{var}(q)$ to denote the set of variables in q.

A match of q in $\mathcal{I}$ is a mapping $\pi : \mathsf{var}(q) \rightarrow \Delta^{\mathcal{I}}$ such that

- $\pi(x) \in A^{\mathcal{I}}$ for all concept atoms $A(x)$ in q, and
- $(\pi(x), \pi(y)) \in r^{\mathcal{I}}$ for all role atoms $r(x, y)$ in q.

Let $\vec{x} = x_1, \ldots, x_k$ be the answer variables in q and
$\vec{a} = a_1, \ldots, a_k$ individual names from $\mathbf{I}$.

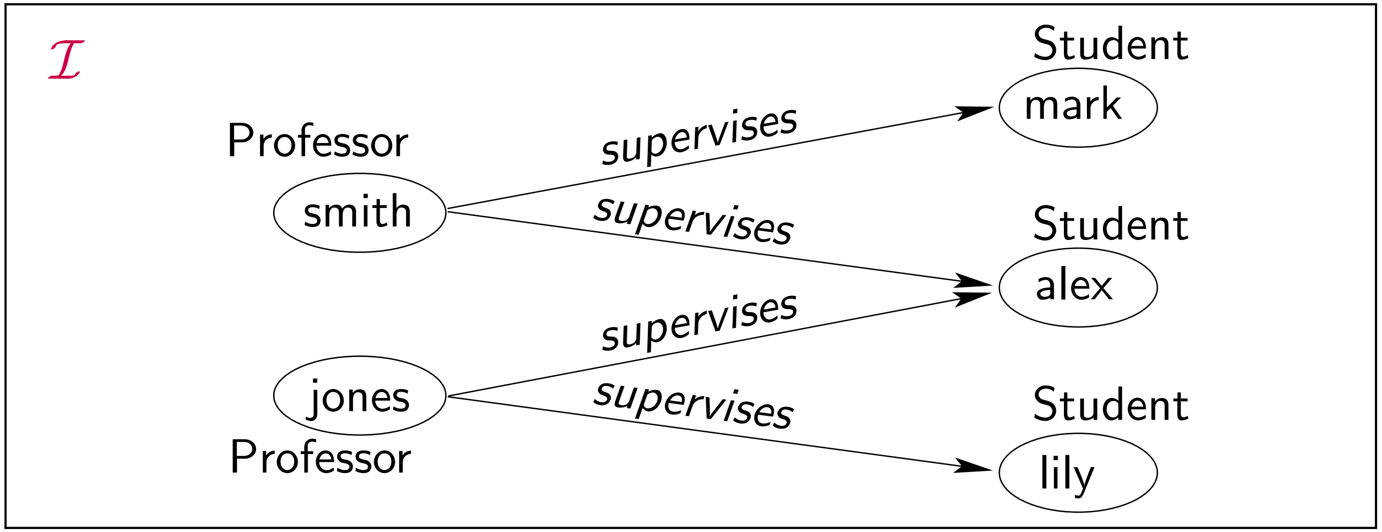
We call the match π of q in $\mathcal{I}$ an $\vec{a}$-match if $\pi(x_i) = a_i^{\mathcal{I}}$ for $1 \leq i \leq k$.

Lemma 7.4

$\mathsf{ans}(q, \mathcal{I}) = \{\vec{a} \mid$ there is an $\vec{a}$-match of q in $\mathcal{I}\}$

Conjunctive queries

examples



$$q_1(x_1, x_2) = \text{Professor}(\underline{x_1}) \wedge \text{supervises}(\underline{x_1}, \underline{x_2}) \wedge \text{Student}(\underline{x_2})$$

$$q_2(x) = \exists y\ (\text{Professor}(y) \wedge \text{supervises}(y, \underline{x}) \wedge \text{Student}(\underline{x}))$$

$$q_3(x_1, x_2) = \exists y\ (\text{Professor}(y) \wedge \text{supervises}(y, \underline{x_1}) \wedge \text{supervises}(y, \underline{x_2}) \wedge \text{Student}(\underline{x_1}) \wedge \text{Student}(\underline{x_2}))$$

Dresden

Complexity of conjunctive query answering

We consider the following decision problem:

Definition 7.5 (query entailment)

Let q be a conjunctive query of arity k, $\mathcal{I}$ an interpretation and $\vec{a} = a_1, \ldots, a_k$ a tuple of individuals.

We say that $\mathcal{I}$ entails $q(\vec{a})$ (and write $\mathcal{I} \models q(\vec{a})$) if $\vec{a} \in \mathsf{ans}(q, \mathcal{I})$.

If $k = 0$, then we call q a Boolean query and simply write $\mathcal{I} \models q$.

Proposition 7.6

The query entailment problem for conjunctive queries is NP-complete.

Proof:

In NP:

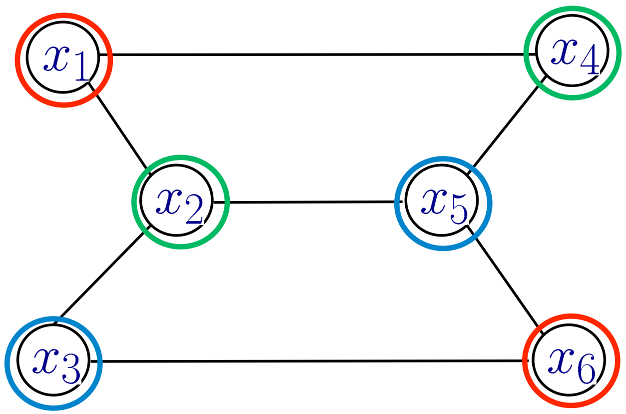
guess a mapping $\pi : \mathsf{var}(q) \rightarrow \Delta^{\mathcal{I}}$ and test whether it is an $\vec{a}$-match.

NP-hard:

reduction of 3-colorability

Complexity

3-colorability: a well-known NP-complete problem



The (undirected) graph $G = (V, E)$ is 3-colorable if there is a mapping $c : V \to \{\textit{red}, \textit{blue}, \textit{green}\}$ such that $\{u, v\} \in V$ implies $c(u) \neq c(v)$.

Conjunctive query q:

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

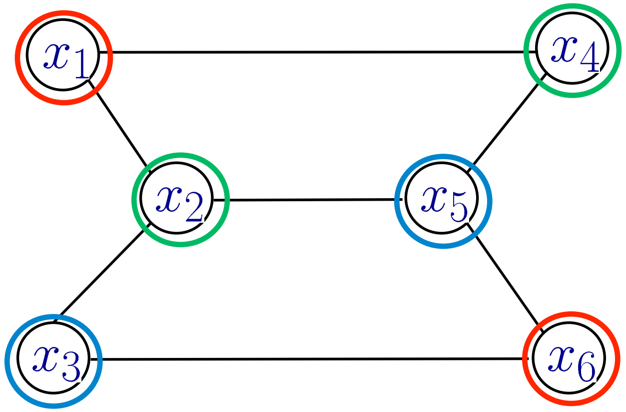
Interpretation $\mathcal{I}$:

$$\Delta^{\mathcal{I}} = \{\textit{red}, \textit{blue}, \textit{green}\}$$

$$E^{\mathcal{I}} = \{(\textit{red}, \textit{blue}), (\textit{blue}, \textit{red})\\ (\textit{red}, \textit{green}), (\textit{green}, \textit{red})\\ (\textit{green}, \textit{blue}), (\textit{blue}, \textit{green})\}$$

Complexity

3-colorability: a well-known NP-complete problem



The (undirected) graph $G = (V, E)$ is 3-colorable if there is a mapping $c : V \to \{\mathit{red}, \mathit{blue}, \mathit{green}\}$ such that $\{u, v\} \in V$ implies $c(u) \neq c(v)$.

Conjunctive query q: general definition

$$\exists v_1, \ldots, v_k. \bigwedge_{\{u,v\} \in E} E(u, v)$$

$\mathcal{I} \models q$ iff G is 3-colorable

Interpretation $\mathcal{I}$:

$$\Delta^{\mathcal{I}} = \{\mathit{red}, \mathit{blue}, \mathit{green}\}$$

$$E^{\mathcal{I}} = \{(\mathit{red}, \mathit{blue}), (\mathit{blue}, \mathit{red}) \\ (\mathit{red}, \mathit{green}), (\mathit{green}, \mathit{red}) \\ (\mathit{green}, \mathit{blue}), (\mathit{blue}, \mathit{green})\}$$

Dresden

data complexity

In practice:

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

Why doesn't this contradict the NP-hardness result?

In practice:

- the size of the data is very large,
- whereas the size of the query is small

In contrast, in our reduction the query had the size of the graph, and the data had constant size.

Data complexity

Measure the complexity in the size of the data only, and and assume that the query has constant size.

Complexity

data complexity

Proposition 7.7

The query entailment problem for conjunctive queries is in P w.r.t. data complexity.

Proof:

Generate all mappings $\pi : \mathsf{var}(q) \to \Delta^{\mathcal{I}}$ and test whether any of them is an $\vec{a}$-match.

There are $|\Delta^{\mathcal{I}}|^{|\mathsf{var}(q)|}$ such mappings. Polynomially many!

size of data (→ $|\Delta^{\mathcal{I}}|$) constant (→ $|\mathsf{var}(q)|$)

Complexity

data complexity

Proposition 7.7

The query entailment problem for conjunctive queries is in P w.r.t. data complexity.

One can even show that the query entailment problem for FO queries (and thus also conjunctive queries) belongs to a complexity class strictly contained in P w.r.t. data complexity.

Theorem 7.8

The query entailment problem for FO queries is in AC^0 w.r.t. data complexity.

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

Ontology-mediated query answering

In OMQA we consider:

- a TBox $\mathcal{T}$ that represents background knowledge,
- an ABox $\mathcal{A}$ that gives an incomplete description of the data,
- a conjunctive query q.

What are the actual data (i.e., the interpretation $\mathcal{I}$) is not known,
all we know is that they are consistent with $\mathcal{T}$ and $\mathcal{A}$,
i.e., $\mathcal{I}$ is a model of $\mathcal{T} \cup \mathcal{A}$.

We want to find answers to q that are true for all possible data,
i.e., for all models of $\mathcal{T} \cup \mathcal{A}$:

Certain Answers

Dresden

Certain answers in the OMQA setting

Definition 7.9 (certain answer)

Let $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ be a knowledge base.

Then $\vec{a}$ is a certain answer to q on $\mathcal{K}$ if

- all individual names from $\vec{a}$ occur in $\mathcal{A}$ and
- $\vec{a} \in \mathsf{ans}(q, \mathcal{I})$ for every model $\mathcal{I}$ of $\mathcal{K}$.

We use $\mathsf{cert}(q, \mathcal{K})$ to denote the set of all certain answers to q on $\mathcal{K}$, i.e.,

$$\mathsf{cert}(q, \mathcal{K}) = \bigcap_{\mathcal{I} \text{ model of } \mathcal{K}} \mathsf{ans}(q, \mathcal{I}).$$

Note:

$\vec{a} \in \mathsf{cert}(q, \mathcal{K})$ iff $\mathcal{T} \cup \mathcal{A} \models q(\vec{a})$

Dresden

Certain answers in the OMQA setting

Example

$$\mathcal{T} = \{\mathsf{Student} \sqsubseteq \exists \mathsf{supervises}^-.\mathsf{Professor}\}$$

$$\mathcal{A} = \{\mathsf{smith} : \mathsf{Professor}, \mathsf{mark} : \mathsf{Student}, \mathsf{alex} : \mathsf{Student}, \mathsf{lily} : \mathsf{Student}, (\mathsf{smith}, \mathsf{mark}) : \mathsf{supervises}, (\mathsf{smith}, \mathsf{alex}) : \mathsf{supervises}\}$$

$$q_1(x_1, x_2) = \mathsf{Professor}(\underline{x_1}) \wedge \mathsf{supervises}(\underline{x_1}, \underline{x_2}) \wedge \mathsf{Student}(\underline{x_2})$$

$$q_2(x) = \exists y\ (\mathsf{Professor}(y) \wedge \mathsf{supervises}(y, \underline{x}) \wedge \mathsf{Student}(\underline{x}))$$

$$q_3(x_1, x_2) = \exists y\ (\mathsf{Professor}(y) \wedge \mathsf{supervises}(y, \underline{x_1}) \wedge \mathsf{supervises}(y, \underline{x_2}) \wedge \mathsf{Student}(\underline{x_1}) \wedge \mathsf{Student}(\underline{x_2}))$$

Complexity of OMQA

In the context of OMQA, query entailment is redefined as follows:

Definition 7.10 (OMQA query entailment)

Let q be a conjunctive query of arity k, $\mathcal{K}$ a knowledge base and $\vec{a} = a_1, \ldots, a_k$ a tuple of individuals occurring in $\mathcal{K}$.

We say that $\mathcal{K}$ entails $q(\vec{a})$ (and write $\mathcal{K} \models q(\vec{a})$) if $\vec{a} \in \mathsf{cert}(q, \mathcal{K})$.

If $k = 0$, then we simply write $\mathcal{K} \models q$.

Data complexity

Consider only simple ABoxes, whose assertions are of the form $a : A$ and $(a, b) : r$ where $A \in \mathbf{C}$ and $r \in \mathbf{R}$.

Measure the complexity in the size of the ABox only, and assume that the TBox and the query have constant size.

Dresden

Complexity

of OMQA

The complexity of OMQA query entailment of course depends on which query language and which DL for formulating the KB are used.

Query language

We consider only conjunctive queries.

In fact, for FO queries, OMQA query entailment would be undecidable.

Blackboard

Description Logics

The data complexity of OMQA query entailment may vary considerably:

$\mathcal{ALC}$: coNP-complete	We will show coNP-hardness.
$\mathcal{EL}$: P-complete	We will show P-hardness.
DL-Lite: AC^0	We will sketch how to show in AC^0.

Dresden

data complexity of OMQA in $\mathcal{ALC}$

Proposition 7.11

In $\mathcal{ALC}$, the query entailment problem for conjunctive queries is coNP-hard w.r.t. data complexity.

Proof: by reduction of non-3-colorability

The TBox and the query are constant, i.e., they do not depend on the input graph.

The input graph $G = (V, E)$ is translated into the ABox

$$
\begin{array}{rcl}
\mathcal{T} & = & \{ \quad \top \sqsubseteq R \sqcup G \sqcup B \\
& & R \sqcap \exists r.R \sqsubseteq D \\
& & G \sqcap \exists r.G \sqsubseteq D \\
& & B \sqcap \exists r.B \sqsubseteq D \quad \} \\
q & = & \exists x\, D(x)
\end{array}
$$

$$\mathcal{A}_G := \{(u, v) : r \mid \{u, v\} \in E\}$$

We have $(\mathcal{T}, \mathcal{A}_G) \models q$ iff G is not 3-colorable.

Blackboard

Complexity

data complexity of OMQA in $\mathcal{EL}$

Proposition 7.12

In $\mathcal{EL}$, the query entailment problem for conjunctive queries is P-hard w.r.t. data complexity.

Proof: by LogSpace-reduction of path system accessibility

A path system is of the form $P = (N, E, S, t)$ where

- N is a finite set of nodes,
- $E \subseteq N \times N \times N$ is an accessibility relation (we call its elements edges),
- $S \subseteq N$ is a set of source nodes,
- and $t \in N$ is a terminal node.

The set of accessible nodes of P is the smallest set of nodes such that

- every element of S is accessible,
- if n_1, n_2 are accessible and $(n, n_1, n_2) \in E$, then n is accessible.

Dresden

Complexity

data complexity of OMQA in $\mathcal{EL}$

Path system accessibility:

Given: a path system $P = (N, E, S, t)$

Question: is t accessible?

The reduction:

$$\mathcal{T} = \{\exists P_1.A \sqsubseteq B_1,\ \exists P_2.A \sqsubseteq B_2,\ B_1 \sqcap B_2 \sqsubseteq A,\ \exists P_3.A \sqsubseteq A\}$$

$$q = A(x)$$

$$\mathcal{A} = \{A(n) \mid n \in S\} \cup \{P_1(e,j),\ P_2(e,k),\ P_3(n,e) \mid e = (n,j,k) \in E\}$$

We have $(\mathcal{T}, \mathcal{A}) \models A(t)$ iff t is accessible in P.

Blackboard

Dresden

Ontology-mediated query answering

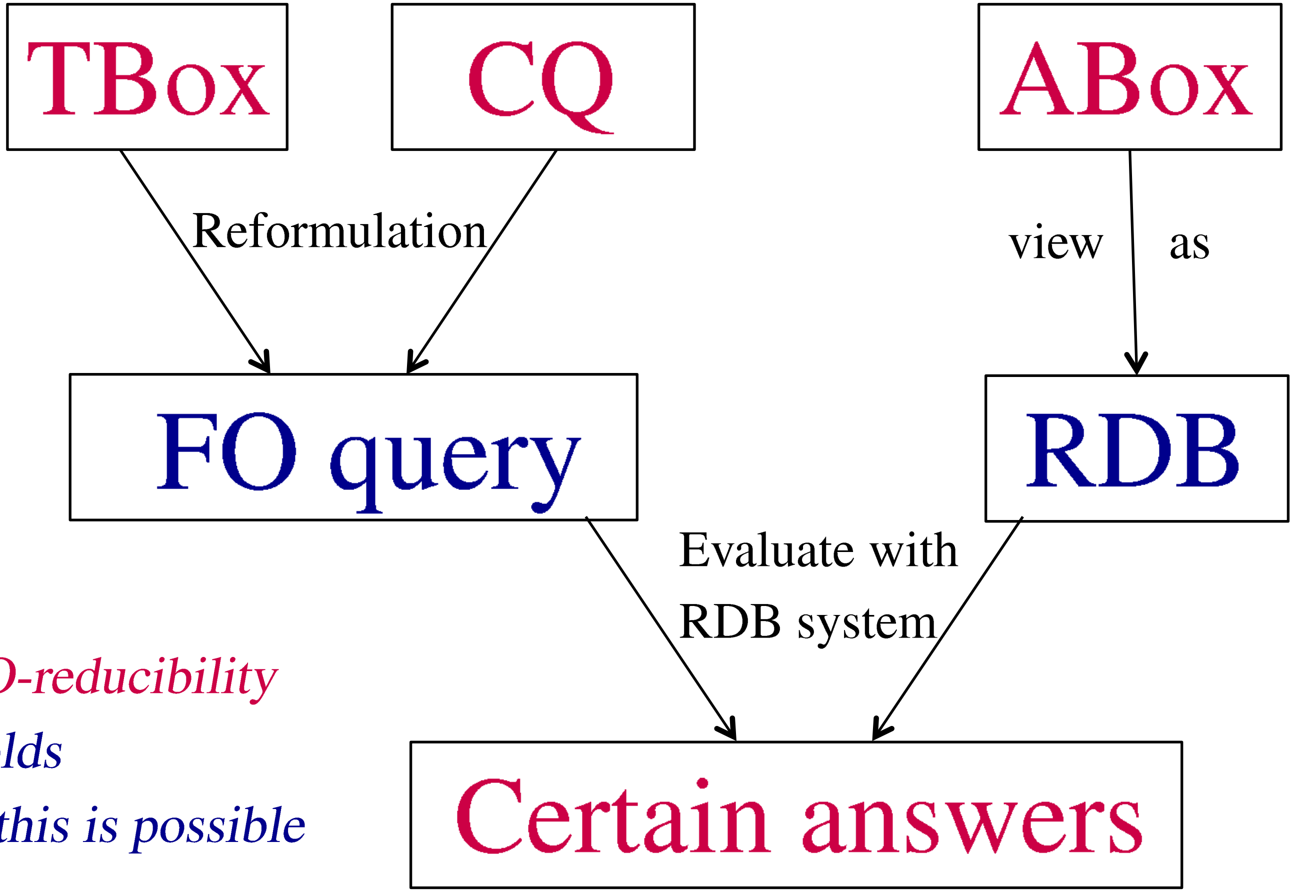
In order to deal with very large ABoxes,
tractability (i.e., in P) is not sufficient.

Goal

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

Dresden

OMQA using relational DB technology



FO-reducibility
holds
if this is possible

Ontology-mediated query answering

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

Goal

Find DLs for which FO-reducibility holds.

⇒ the DL-Lite family

Dresden

DL-Lite$_{core}$

the basic member of the DL-Lite family

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

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

$LINDA : Woman$

$(LINDA, JAMES) : has\_child$

$PAUL : Beatle$

$(PAUL, JAMES) : has\_child$

Ontology-mediated query answering in DL-Lite$_{core}$

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

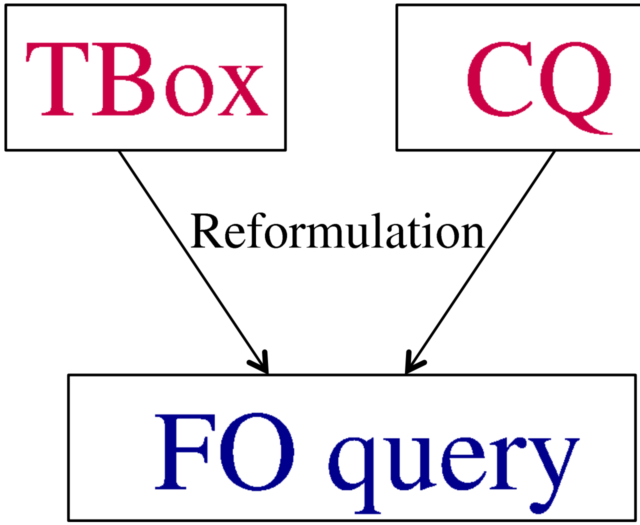
↑ answer variable

certain answer: $(LINDA)$

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

FO-reducibility

of DL-Lite$_{core}$



Query reformulation generates a disjunction 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.

FO-reducibility

of DL-Lite$_{core}$

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

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

TBox

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

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

Dresden

FO-reducibility

of DL-Lite$_{core}$

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

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

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

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

FO-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

FO-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

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

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

FO-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: $(LINDA)$

Dresden

FO-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

FO-reducibility

for the DL-Lite family of DLs

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

additional role inclusion axioms:

$r_1 \sqsubseteq r_2$

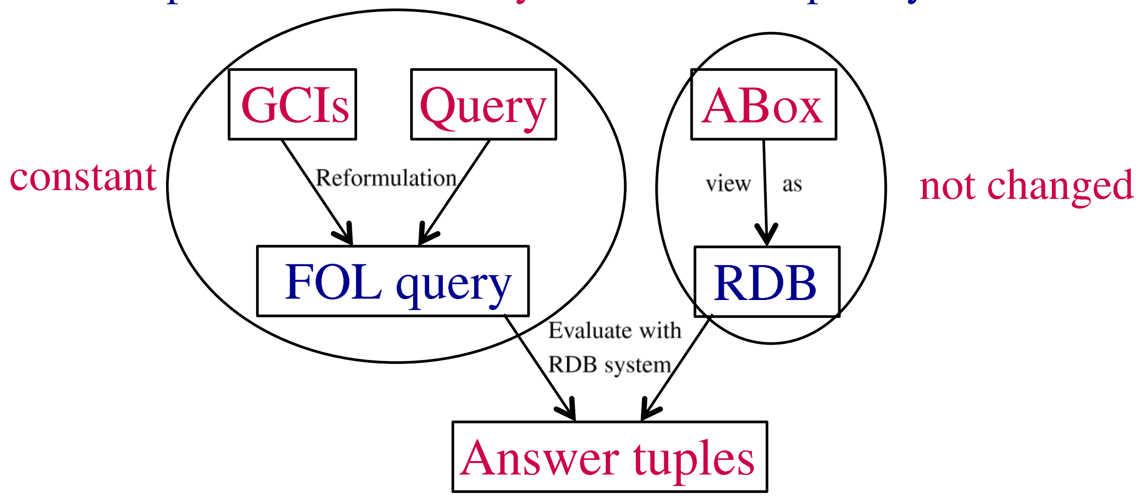
$r_1 \sqsubseteq \neg r_2$

additional functionality axioms:

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

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

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



Ontology-mediated query answering in $\mathcal{EL}$

- Computing certain answers w.r.t. $\mathcal{EL}$-TBoxes is polynomial w.r.t. data complexity.
- However it is also P-hard, and thus not in AC^0.
- Thus, query answering in $\mathcal{EL}$ is not FO-reducible.

Can we still use RDB technology for query evaluation?

- Yes, but one needs to rewrite into Datalog.
- Datalog-rewritability even holds for $\mathcal{ELI}$.

See Section 7.2 in the book.