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.

---

<table>
<thead>
<tr>
<th>Database</th>
<th>finite collection of relations over a finite domain</th>
<th>SQL query describing answer tuples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB = finite interpretation</td>
<td></td>
<td>basically same expressiveness</td>
</tr>
</tbody>
</table>

| 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) |
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 $\bar{x}$ of an FO query $q(\bar{x})$ are called answer variables.

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

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

$$\bar{a} = a_1, \ldots, a_k$$

is an answer to $q$ on $\mathcal{I}$ if $\mathcal{I} \models q[\bar{a}]$

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

$\text{ans}(q, \mathcal{I})$: set of all answers to $q$ in $\mathcal{I}$
**Conjunctive queries** restricted class of FO queries

**Definition 7.2** (conjunctive query)

A conjunctive query (CQ) \( q \) has the form

\[
\exists x_1 \cdots \exists x_k (\alpha_1 \land \cdots \land \alpha_n)
\]

where \( k \geq 0 \), \( n \geq 1 \), \( x_1, \ldots, x_k \) are variables, and each \( \alpha_i \) is a concept atom \( A(x) \) or a role atom \( r(x, y) \) with \( A \in C \), \( r \in R \), 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 \( q \).
Conjunctive queries

\[ q_1(x_1, x_2) = \text{Professor}(x_1) \land \text{supervises}(x_1, x_2) \land \text{Student}(x_2) \]

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

\[ q_2(x) = \exists y \left( \text{Professor}(y) \land \text{supervises}(y, x) \land \text{Student}(x) \right) \]

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

\[ q_3(x_1, x_2) = \exists y \left( \text{Professor}(y) \land \text{supervises}(y, x_1) \land \text{supervises}(y, x_2) \land \text{Student}(x_1) \land \text{Student}(x_2) \right) \]

Returns all pairs of students supervised by the same professor.
**Conjunctive queries**

characterisation of answer tuples

**Definition 7.3** (\(\bar{a}\)-match)

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

A match of \(q\) in \(\mathcal{I}\) is a mapping \(\pi : \text{var}(q) \rightarrow \Delta^\mathcal{I}\) 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 \(\bar{x} = x_1, \ldots, x_k\) be the answer variables in \(q\) and \(\bar{a} = a_1, \ldots, a_k\) individual names from \(\mathcal{I}\).

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

**Lemma 7.4**

\[
\text{ans}(q, \mathcal{I}) = \{\bar{a} \mid \text{there is an } \bar{a}\text{-match of } q \text{ in } \mathcal{I}\}
\]
7.1 Conjunctive Queries and FO Queries

Definition 7.3

Let \( K = (A, T) \) be a knowledge base. Then \(~a\) is a certain answer to \( q \) on \( K \) if all individual names from \(~a\) occur in \( A \) and \(~a\) of \( \text{ans}(q, I) \) for every model \( I \) of \( K \). We use \( \text{cert}(q, K) \) to denote the set of all certain answers to \( q \) on \( K \), that is, \( \text{cert}(q, K) = \bigcap_{I \text{ model of } K} \text{ans}(q, I) \).

As an example, consider the following knowledge base \( K = (T, A) \) formulated in ALCI:

\[
T = \{ \text{Student} \, v9 \, \text{supervises} \, \text{Professor} \}
\]

\[
A = \{ \text{smith} : \text{Professor}, \text{mark} : \text{Student}, \text{alex} : \text{Student}, \text{lily} : \text{Student} \}
\]

Note that the interpretation in Figure 7.1 is a model of this KB. Let us first consider the query \( q_4(x) \) from above. As expected, we have \( \text{cert}(q_4, K) = \{ \text{mark}, \text{alex} \} \). It is easy to find models of \( K \) in which \text{smith} supervises more students than \text{mark} and \text{alex}, but the latter are the only two students on whose supervision by \text{smith} all models are in agreement.

It is illustrating to consider the role of domain elements whose existence is enforced by existential restrictions in the TBox. For the query \( q_2(x) \), we find \( \text{cert}(q_2, K) = \{ \text{mark}, \text{alex}, \text{lily} \} \). Note that \( \text{lily} \) is included because

\[
q_1(x_1, x_2) = \text{Professor}(x_1) \land \text{supervises}(x_1, x_2) \land \text{Student}(x_2)
\]

\[
q_2(x) = \exists y \ (\text{Professor}(y) \land \text{supervises}(y, x) \land \text{Student}(x))
\]

\[
q_3(x_1, x_2) = \exists y \ (\text{Professor}(y) \land \text{supervises}(y, x_1) \land \text{supervises}(y, x_2) \land \text{Student}(x_1) \land \text{Student}(x_2))
\]
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 \( \bar{a} = a_1, \ldots, a_k \) a tuple of individuals.

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

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 \( \pi : \text{var}(q) \rightarrow \Delta^\mathcal{I} \) and
- test whether it is an \( \bar{a} \)-match.

NP-hard:

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

Conjunctive query $q$:

$\exists x_1, x_2, x_3, x_4, x_5, x_6.$

$E(x_1, x_2) \land E(x_2, x_3) \land$

$E(x_1, x_4) \land E(x_2, x_5) \land E(x_3, x_6) \land$

$E(x_4, x_5) \land E(x_5, x_6)$

Interpretation $\mathcal{I}$:

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

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

$(\text{red, green}), (\text{green, red})$

$(\text{green, blue}), (\text{blue, green})\}$
Complexity

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

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

Conjunctive query $q$: general definition

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

Interpretation $\mathcal{I}$:

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

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

$(\text{red, green}), (\text{green, red})$

$(\text{green, blue}), (\text{blue, green})\}$
Complexity

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 assume that the query has constant size.
**Proposition 7.7**

The query entailment problem for conjunctive queries is in $\mathbb{P}$ w.r.t. data complexity.

**Proof:**

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

There are $|\Delta^I|^{\text{var}(q)}$ such mappings. Polynomials many!

size of data constant
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.

$AC^0 \subset LogSpace \subset P$
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
Certain answers in the OMQA setting

**Definition 7.9** (certain answer)

Let $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ be a knowledge base. Then $\bar{a}$ is a certain answer to $q$ on $\mathcal{K}$ if

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

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

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

**Note:**

$\bar{a} \in \text{cert}(q, \mathcal{K})$ iff $\mathcal{T} \cup \mathcal{A} \models q(\bar{a})$
Certain answers in the OMQA setting

Example

\[ \mathcal{T} = \{ \text{Student} \sqsubseteq \exists \text{supervises} \neg . \text{Professor} \} \]
\[ \mathcal{A} = \{ \text{smith} : \text{Professor}, \text{mark} : \text{Student}, \text{alex} : \text{Student}, \text{lily} : \text{Student}, \]
\[ (\text{smith, mark}) : \text{supervises}, (\text{smith, alex}) : \text{supervises} \} \]

\[ q_1(x_1, x_2) = \text{Professor}(x_1) \land \text{supervises}(x_1, x_2) \land \text{Student}(x_2) \]

\[ q_2(x) = \exists y \ (\text{Professor}(y) \land \text{supervises}(y, x) \land \text{Student}(x)) \]

\[ q_3(x_1, x_2) = \exists y \ (\text{Professor}(y) \land \text{supervises}(y, x_1) \land \text{supervises}(y, x_2) \land \]
\[ \text{Student}(x_1) \land \text{Student}(x_2)) \]
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 \( \bar{a} = a_1, \ldots, a_k \) a tuple of individuals occurring in \( \mathcal{K} \).

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

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 \( A \in C \) and \( r \in R \).

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

\[ \mathcal{ALC} : \text{coNP-complete} \quad \text{We will show coNP-hardness.} \]

\[ \mathcal{EL} : \text{P-complete} \quad \text{We will show P-hardness.} \]

\[ \text{DL-Lite: AC}^0 \quad \text{We will sketch how to show in AC}^0. \]
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

$$
\begin{align*}
\mathcal{T} &= \{ & \top & \sqsubseteq R \sqcup G \sqcup B \\
           &     & R \cap \exists r.R & \sqsubseteq D \\
           &     & G \cap \exists r.G & \sqsubseteq D \\
           &     & B \cap \exists r.B & \sqsubseteq D \} \\
q &= \exists x \ D(x)
\end{align*}
$$

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

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

$q = A(x)$

$\mathcal{A} = \{ A(n) \mid n \in S \} \cup$

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

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

Blackboard
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.
OMQA using relational DB technology

TBox \rightarrow \text{Reformulation} \rightarrow \text{FO query} \rightarrow \text{Evaluate with RDB system} \rightarrow \text{Certain answers}

CQ

ABox \rightarrow \text{view as} \rightarrow \text{RDB}

\text{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
**DL-Lite\textsubscript{core}**

the basic member of the DL-Lite family

- concept names: $A$
- basic concepts: $B$
- general concepts: $C$

$B \rightarrow A \lor E \cdot T \lor E^{-1} \cdot T$  

$C \rightarrow B \lor \neg B$

**GCIs**

- $B \subseteq C$
- $\exists \text{has\_child}.T \subseteq \neg \text{Spinster}$
- $\exists \text{has\_child}.T \subseteq \text{Parent}$
- $\text{Parent} \subseteq \text{Human}$
- $\text{Human} \subseteq \exists \text{has\_child}^{-1}.T$

**ABox**

- $A(a)$
- $r(a, b)$
- $\text{LINDA}: \text{Woman}$
- $(\text{LINDA}, \text{JAMES}): \text{has\_child}$
- $\text{PAUL}: \text{Beatle}$
- $(\text{PAUL}, \text{JAMES}): \text{has\_child}$
Ontology-mediated query answering in DL-Lite\textsubscript{core}

\[ \exists y, z_1, z_2. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \land \text{has\_child}(z_2, z_1) \]

answer variable

certain answer: (\textit{LINDA})

<table>
<thead>
<tr>
<th>TBox</th>
<th>ABox</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists \text{has_child}. \top \sqsubseteq \neg \text{Spinsten}$</td>
<td>$\text{LINDA}: \text{Woman}$</td>
</tr>
<tr>
<td>$\exists \text{has_child}. \top \sqsubseteq \text{Parent}$</td>
<td>$(\text{LINDA}, \text{JAMES}) : \text{has_child}$</td>
</tr>
<tr>
<td>$\text{Parent} \sqsubseteq \text{Human}$</td>
<td>$\text{PAUL} : \text{Beatle}$</td>
</tr>
<tr>
<td>$\text{Human} \sqsubseteq \exists \text{has_child}^{-1}. \top$</td>
<td>$(\text{PAUL}, \text{JAMES}) : \text{has_child}$</td>
</tr>
</tbody>
</table>
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\textsubscript{core}

\begin{align*}
\exists y, z_1, z_2. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \land \text{has\_child}(z_2, z_1)
\end{align*}

\begin{align*}
\exists y, z_1. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \land \text{Human}(z_1)
\end{align*}

**TBox**

\begin{align*}
\exists \text{has\_child}. \top \sqsubseteq \neg \text{Spinster} \\
\text{Parent} \sqsubseteq \text{Human} \\
\exists \text{has\_child}. \top \sqsubseteq \text{Parent} \\
\text{Human} \sqsubseteq \exists \text{has\_child}^{-1}. \top
\end{align*}
FO-reducibility of DL-Lite$_{\text{core}}$

\[
\exists y, z_1, z_2. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \land \text{has\_child}(z_2, z_1)
\]

\[
\exists y, z_1. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1)
\]

\[
\exists y, z_1. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Parent}(z_1)
\]

**TBox**

\[
\exists \text{has\_child}. \top \sqsubseteq \neg \text{Spinster} \quad \exists \text{has\_child}. \top \sqsubseteq \text{Parent}
\]

\[
\text{Parent} \sqsubseteq \text{Human} \quad \text{Human} \sqsubseteq \exists \text{has\_child}^{-1}. \top
\]
FO-reducibility of DL-Lite$_{core}$

\[ \exists y, z_1, z_2. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \land \text{has\_child}(z_2, z_1) \]

\[ \exists y, z_1. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \]

\[ \exists y, z_1. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Parent}(z_1) \]

\[ \exists y, z_1, z_3. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{has\_child}(z_1, z_3) \]

TBox

\[ \exists \text{has\_child}. \top \subseteq \neg \text{Spinster} \quad \exists \text{has\_child}. \top \subseteq \text{Parent} \]

\[ \text{Parent} \subseteq \text{Human} \quad \text{Human} \subseteq \exists \text{has\_child}^{-1}. \top \]
FO-reducibility of DL-Lite\textsubscript{core}

\[ \exists y, z_1, z_2. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \land \text{has\_child}(z_2, z_1) \]

\[ \exists y, z_1. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \]

\[ \exists y, z_1. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Parent}(z_1) \]

\[ \exists y, z_1, z_3. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{has\_child}(z_1, z_3) \]

\[ \text{ABox} \]

\begin{align*}
\text{LINDA: Woman} & \quad (\text{LINDA, JAMES}): \text{has\_child} \\
\text{PAUL: Beatle} & \quad (\text{PAUL, JAMES}): \text{has\_child}
\end{align*}

\[ \text{TBox} \]

\begin{align*}
\exists \text{has\_child}. \top & \subseteq \neg \text{Spinster} & \exists \text{has\_child}. \top & \subseteq \text{Parent} \\
\text{Parent} & \subseteq \text{Human} & \text{Human} & \subseteq \exists \text{has\_child}^{-1}. \top
\end{align*}
FO-reducibility of DL-Lite$_{core}$

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

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

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

$\exists y, z_1, z_3. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{has\_child}(z_1, z_3)$

**RDB**

<table>
<thead>
<tr>
<th>Woman</th>
<th>has_child</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINDA</td>
<td>LINDA, JAMES</td>
</tr>
<tr>
<td>PAUL</td>
<td>PAUL, JAMES</td>
</tr>
</tbody>
</table>

**answer tuple:** (LINDA)
FO-reducibility of DL-Lite$_{\text{core}}$

Some subtleties

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

\[ \exists y, z_1, z_2. \text{Woman}(x) \land \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \land \text{Human}(z_1) \land \text{has\_child}(z_2, z_1) \]

\[ \text{Human} \sqsubseteq \exists \text{has\_child}^{-1}. \top \]

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

\[ \exists y, z_1. \text{has\_child}(x, y) \land \text{has\_child}(z_1, y) \]

\[ \text{Parent} \sqsubseteq \exists \text{has\_child}. \top \]

Unification replaces \( z_1 \) by \( x \):

\[ \exists y. \text{has\_child}(x, y) \]

\[ \text{Parent}(x) \]
FO-reducibility for the DL-Lite family of DLs

- DL-Lite$_{core}$ and its extensions DL-Lite$_R$ and DL-Lite$_E$ are FO-reducible.
  
  **additional role inclusion axioms:**
  
  \[ r_1 \sqsubseteq r_2 \]
  \[ r_1 \sqsubseteq \neg r_2 \]

  **additional functionality axioms:**
  
  \[ T \sqsubseteq (\leq 1 r) \]
  \[ T \sqsubseteq (\leq 1 r^{-1}) \]

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

Diagram:
- GCI
- Query
- ABox
- FOL query
- RDB
- Evaluate with RDB system
- View as
- Answer tuples

Constant

Reformulation
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 $\text{AC}^0$.
- 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 $\mathcal{ELT}$.

*See Section 7.2 in the book.*