# Optimizing Ontology-Based Data Access

*Diego Calvanese* and Mariano Rodriguez-Muro

KRDB Research Centre for Knowledge and Data
Free University of Bozen-Bolzano


ISO 15926 and Semantic Technologies 2012
6–7 September 2012, Sogndal, Norway

Optique™

# Use case: Statoil Exploration

*Experts in geology and geophysics develop stratigraphic models of unexplored areas on the basis of data acquired from previous operations at nearby geographical locations.*


Facts:

- 1,000 TB of relational data
- using diverse schemata
- spread over 2,000 tables, over multiple individual data bases

Data Access for Exploration:

- 900 experts in Statoil Exploration.
- up to 4 days for new data access queries, requiring assistance from IT-experts.
- 30–70% of time spent on data gathering.

Optique unibz.it

# Example 2: Siemens Energy Services

*Runs service centers for power plants, each responsible for remote monitoring and diagnostics of many thousands of gas/steam turbines and associated components. When informed about potential problems, diagnosis engineers access a variety of raw and processed data.*


Facts:

- several TB of time-stamped sensor data
- several GB of event data ("alarm triggered at time T")
- data grows at 30GB per day (sensor data rate 1Hz–1kHz)

Service Requests:

- over 50 service centers worldwide
- 1,000 service requests per center per year
- 80% of time per request used on data gathering

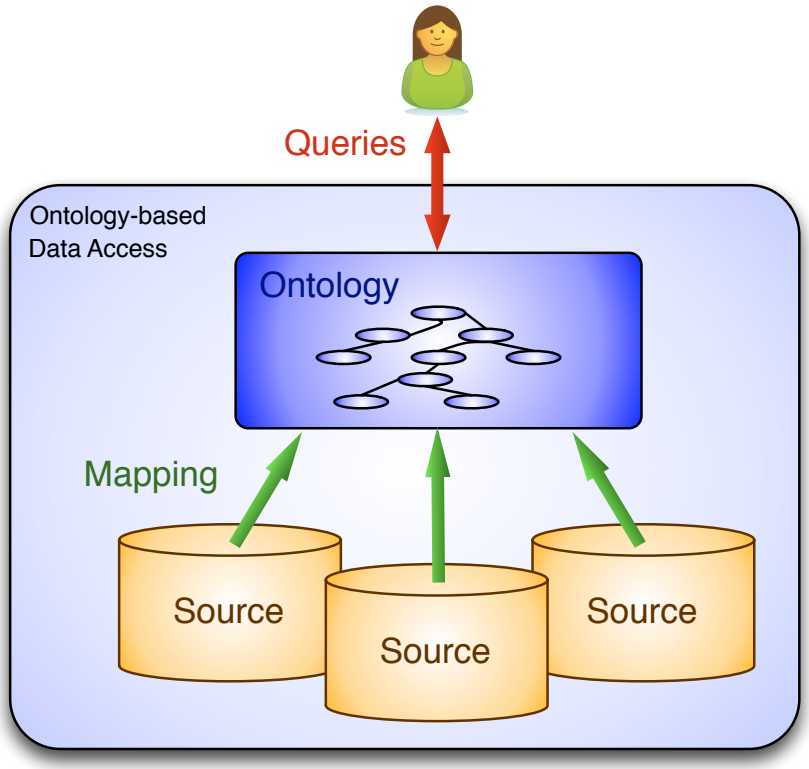
Optique unibz.it

# Proposed solution: Ontology-based Data Access

Manage data adopting principles and techniques studied in **Knowledge Representation** in Artificial Intelligence.

- Based on formalisms grounded in logic, with well understood semantics and computational properties.
- Provide a conceptual, high level representation of the domain of interest in terms of an **ontology**.
- Do **not migrate the data** but leave it in the sources.
- **Map** the ontology to the data sources.
- Specify all information requests to the data in terms of the ontology.
- Use the inference services of the OBDA system to translate the requests into queries to the data sources.

# Ontology-based data access: Architecture




Based on three main components:

- Ontology: provides a unified, conceptual view of the managed information.
- Data source(s): are external and independent (possibly multiple and heterogeneous).
- Mappings: semantically link data at the sources with the ontology.

Optique unibz.it

# Ontology-based data access: Formalization

An **ontology-based access system** is a triple $\mathcal{O} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$, where:

- $\mathcal{T}$ is the intensional level of an ontology.
  We consider ontologies formalized in description logics (DLs), hence the intensional level is a DL TBox.
- $\mathcal{S}$ is a (federated) relational database representing the sources;
- $\mathcal{M}$ is a set of mapping assertions, each one of the form

$$\Phi(\vec{x}) \rightsquigarrow \Psi(\vec{x})$$

  where
  - $\Phi(\vec{x})$ is a FOL query over $\mathcal{S}$, returning tuples of values for $\vec{x}$
  - $\Psi(\vec{x})$ is a FOL query over $\mathcal{T}$ whose free variables are from $\vec{x}$.

# Ontology-based data access: Semantics

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation of the TBox $\mathcal{T}$.

## Def.: **Semantics of an OBDA system**

$\mathcal{I}$ is a **model** of $\mathcal{O} = \langle \mathcal{T}, \mathcal{S}, \mathcal{M} \rangle$ if:

- $\mathcal{I}$ is a FOL model of $\mathcal{T}$, and
- $\mathcal{I}$ satisfies $\mathcal{M}$ w.r.t. $\mathcal{S}$, i.e., satisfies every assertion in $\mathcal{M}$ w.r.t. $\mathcal{S}$.

## Def.: **Semantics of mappings**

We say that $\mathcal{I}$ **satisfies** $\Phi(\vec{x}) \rightsquigarrow \Psi(\vec{x})$ w.r.t. a database $\mathcal{S}$, if the FOL sentence

$$\forall \vec{x}. \, \Phi(\vec{x}) \rightarrow \Psi(\vec{x})$$

is true in $\mathcal{I} \cup \mathcal{S}$.

Optique unibz.it

# Instantiating the framework

1. Which is the "right" **query language**?
2. Which is the "right" **ontology language**?
3. Which is the "right" **mapping language**?

The choices that we make have to take into account the tradeoff between expressive power and efficiency of inference/query answering.

We are in a setting where we want to access large amounts of data, so **efficiency w.r.t. the data** plays an important role.

# Outline

# Use of mappings

In an OBDA system $\mathcal{O} = \langle \mathcal{T}, \mathcal{M}, \mathcal{S} \rangle$, the mapping $\mathcal{M}$ encodes how the data in $\mathcal{S}$ should be used to populate $\mathcal{O}$.

*Observe:*

- The data sources $\mathcal{S}$ and the mapping $\mathcal{M}$ define a virtual ABox $\mathcal{V} = \mathcal{M}(\mathcal{S})$, and queries are answered wrt $\mathcal{T}$ and $\mathcal{V}$.
- We do not really materialize the data of $\mathcal{V}$ (that's why it is called virtual).
- Instead, the intensional information in $\mathcal{T}$ and $\mathcal{M}$ is used to translate (i.e., rewrite and unfold) queries over $\mathcal{T}$ into queries formulated over $\mathcal{S}$.

# The impedance mismatch problem

We have an **impedance mismatch** problem

- In relational databases, information is represented as tuples of values.
- In ontologies, information is represented using both objects and values ...
  - ... with objects playing the main role, ...
  - ... and values a subsidiary role as fillers of object's attributes.

Proposed solution:

- We use **constructors to create objects** of the ontology from tuples of values in the DB.
- The constructors are modeled through Skolem functions in the query over the ontology part of the mapping:

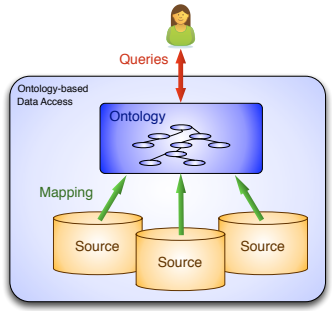
$$\Phi(\vec{x}) \rightsquigarrow \Psi(\vec{\mathbf{f}}, \vec{x})$$

- Techniques from partial evaluation of logic programs are adapted for the unfolding of queries over $\mathcal{T}$ into queries over $\mathcal{S}$ using $\mathcal{M}$.

Optique unibz.it

# Outline

# Incomplete information

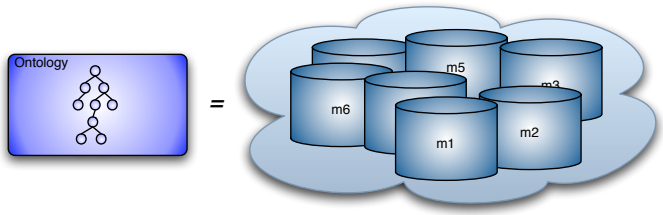



We are in a setting of **incomplete information**!!!

Incompleteness introduced:

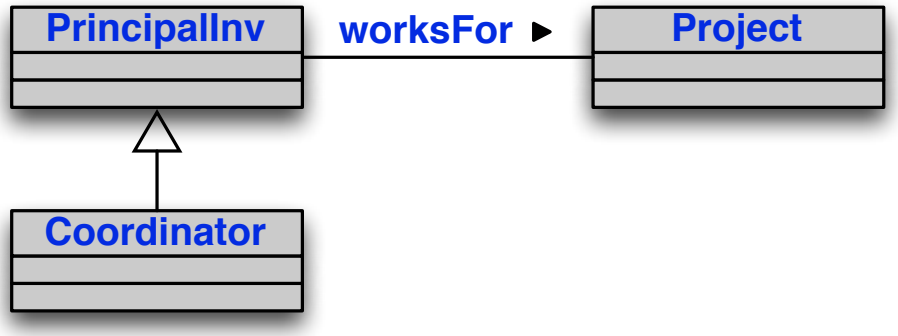
- by data source(s), in general assumed to be incomplete;
- by domain constraints encoded in the ontology.

**Plus:** Ontologies are logical theories, and hence perfectly suited to deal with incomplete information!




**Minus:** Query answering amounts to **logical inference**, and hence is significantly more challenging.

# Incomplete information – Example




Assume for simplicity that each table in the underlying database is mapped directly to (at most) one ontology concept/relationship.

But the database tables may be **incompletely specified**, or even missing for some concepts/relationships.

DB: Coordinator $\supseteq$ { arild, franz }
Project $\supseteq$ { optique, seals }
worksFor $\supseteq$ { (arild,optique), (ian,seals) }

Query: $q(x) \leftarrow \text{PrincipalInv}(x)$

Answer: { arild, franz, ian }

# Query answering

## Certain answers

Query answering amounts to finding the **certain answers** $cert(q, \mathcal{O})$ to a query $q(\vec{x})$, i.e., those answers that hold in all models of the OBDA system $\mathcal{O}$.

Two borderline cases for the language to use for querying ontologies:

1. Use the ontology language as query language.
   - Ontology languages are tailored for capturing intensional relationships.
   - They are quite **poor as query languages**.
2. Full SQL (or equivalently, first-order logic).
   - Problem: in the presence of incomplete information, query answering becomes **undecidable** (FOL validity).

A good tradeoff is to use **conjunctive queries** (CQs) or unions of CQs (UCQs), corresponding to SQL/relational algebra **(union) select-project-join queries**.

# Complexity of conjunctive query answering in DLs

Studied extensively for various ontology languages:

| | Combined complexity | Data complexity |
|---|---|---|
| Plain databases | NP-complete | in $AC^0$ (1) |
| Expressive DLs | $\geq$ 2ExpTime (2) | coNP-hard (3) |

(1) This is what we need to scale with the data.

(2) Hardness by [Lutz, 2008; Eiter *et al.*, 2009].
Tight upper bounds obtained for a variety of expressive DLs [Calvanese *et al.*, 1998; Levy and Rousset, 1998; Calvanese *et al.*, 2007c; Calvanese *et al.*, 2008c; Glimm *et al.*, 2008b; Glimm *et al.*, 2008a; Lutz, 2008; Eiter *et al.*, 2008].

(3) Already for an ontology with a single axiom involving disjunction.
However, the complexity does not increase even for very expressive DLs [Ortiz *et al.*, 2006; Ortiz *et al.*, 2008; Glimm *et al.*, 2008a].

# Challenges in OBDA

## Challenges

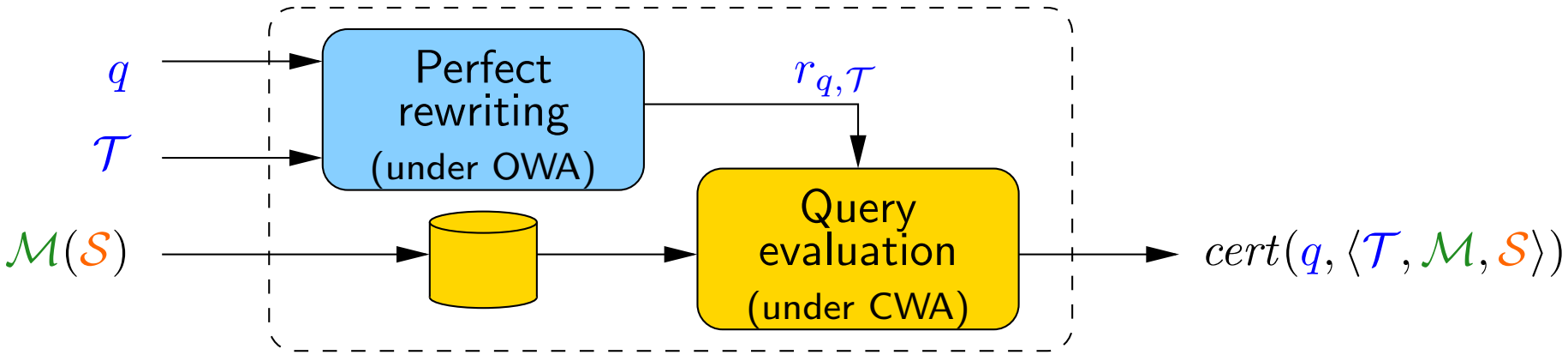
- Can we find interesting ontology languages for which query answering in OBDA can be done efficiently (i.e., in $AC^0$)?
- If yes, can we leverage relational database technology for query answering in OBDA?

# Logical inference for query answering


To be able to deal with data efficiently, we need to separate the contribution of the data $\mathcal{S}$ (accessed via the mapping $\mathcal{M}$) from the contribution of $q$ and $\mathcal{O}$.

$\rightsquigarrow$ Query answering by **query rewriting**.

# Query answering by rewriting




Query answering can always be thought as done in two phases:

1. **Perfect rewriting**: produce from $q$ and the ontology TBox $\mathcal{T}$ a new query $r_{q,\mathcal{T}}$ (called the perfect rewriting of $q$ w.r.t. $\mathcal{T}$).
2. **Query evaluation**: evaluate $r_{q,\mathcal{T}}$ over $\mathcal{M}(\mathcal{S})$ seen as a complete database (and without considering $\mathcal{T}$).
   $\rightsquigarrow$ Produces $cert(q, \langle \mathcal{T}, \mathcal{M}, \mathcal{S} \rangle)$.

We choose the ontology language in such a way that query answering is **FOL-rewritable**, i.e., the perfect rewriting $r_{q,\mathcal{T}}$ can be expressed in FOL (i.e., SQL), hence can be evaluated by a relational DBMS.

Optique unibz.it

# Outline

# The *DL-Lite* family

- A family of DLs optimized according to the tradeoff between expressive power and **complexity** of query answering, with emphasis on **data**.
  - The same complexity as relational databases.
  - In fact, **query answering is FOL-rewritable** and hence can be delegated to a relational DB engine.
  - The DLs of the *DL-Lite* family are essentially the maximally expressive DLs enjoying these nice computational properties.
- Nevertheless they have the "right" expressive power: capture the essential features of conceptual modeling formalisms.

**DL-Lite** provides robust foundations for Ontology-Based Data Access.

*Note:* The *DL-Lite* family is at the basis of the OWL 2 QL profile of the W3C standard Web Ontology Language OWL.

# *DL-Lite* ontologies (essential features)

Concept and role language:

- Roles $R$: either atomic: $P$
  or an inverse role: $P^-$
- Concepts $C$: either atomic: $A$
  or the projection of a role on one component: $\exists P$, $\exists P^-$

TBox assertions:

| | | | |
|---|---|---|---|
| Role inclusion: | $R_1 \sqsubseteq R_2$ | Concept inclusion: | $C_1 \sqsubseteq C_2$ |
| Role disjointness: | $R_1 \sqsubseteq \neg R_2$ | Concept disjointness: | $C_1 \sqsubseteq \neg C_2$ |
| Role functionality: | (**funct** $Q$) | | |

ABox assertions: $A(c)$, $P(c_1, c_2)$, with $c_1$, $c_2$ constants

*Note:* *DL-Lite* distinguishes also between abstract objects and data values (ignored here).

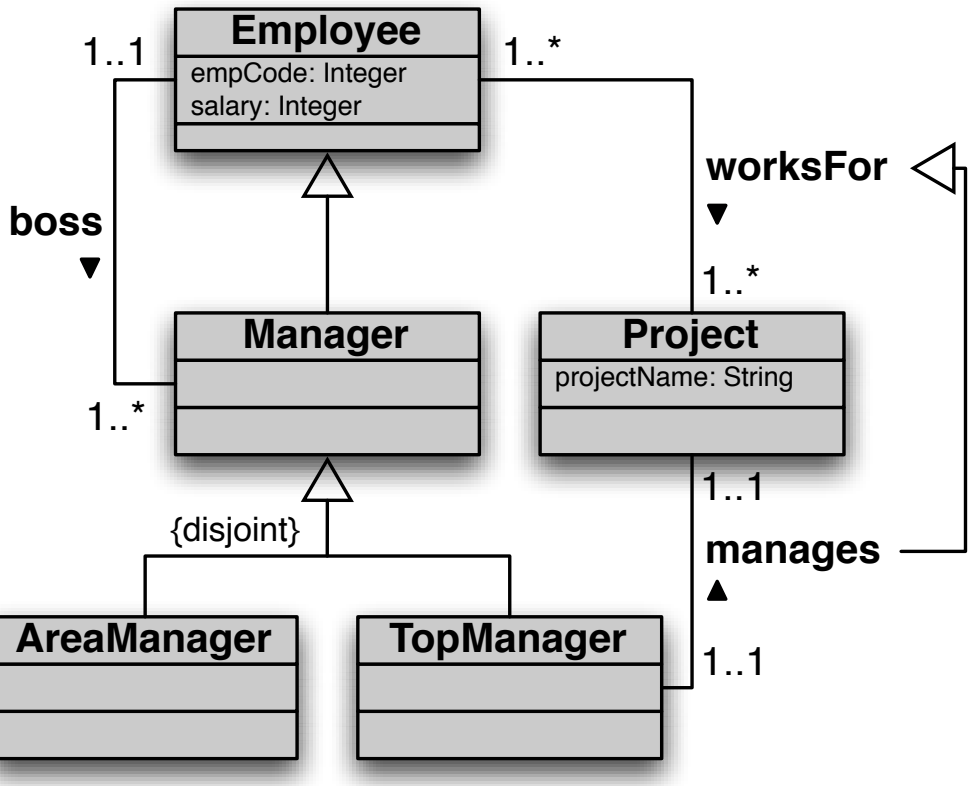
Optique unibz.it

# *DL-Lite* captures conceptual modeling formalisms

| Modeling construct | *DL-Lite* | FOL formalization |
|---|---|---|
| ISA on classes | $A_1 \sqsubseteq A_2$ | $\forall x(A_1(x) \rightarrow A_2(x))$ |
| Disjointness of classes | $A_1 \sqsubseteq \neg A_2$ | $\forall x(A_1(x) \rightarrow \neg A_2(x))$ |
| Domain of relations | $\exists P \sqsubseteq A_1$ | $\forall x(\exists y(P(x,y)) \rightarrow A_1(x))$ |
| Range of relations | $\exists P^- \sqsubseteq A_2$ | $\forall x(\exists y(P(y,x)) \rightarrow A_2(x))$ |
| Mandatory participation *(min card = 1)* | $A_1 \sqsubseteq \exists P$<br>$A_2 \sqsubseteq \exists P^-$ | $\forall x(A_1(x) \rightarrow \exists y(P(x,y)))$<br>$\forall x(A_2(x) \rightarrow \exists y(P(y,x)))$ |
| Functionality *(max card = 1)* | (**funct** $P$)<br>(**funct** $P^-$) | $\forall x,y,y'(P(x,y) \wedge P(x,y') \rightarrow y = y')$<br>$\forall x,x',y(P(x,y) \wedge P(x',y) \rightarrow x = x')$ |
| ISA on relations | $R_1 \sqsubseteq R_2$ | $\forall x,y(R_1(x,y) \rightarrow R_2(x,y))$ |
| Disjointness of relations | $R_1 \sqsubseteq \neg R_2$ | $\forall x,y(R_1(x,y) \rightarrow \neg R_2(x,y))$ |
| ... | ... | ... |

Optique unibz.it

# Capturing UML class diagrams/ER schemas in *DL-Lite*




$$
\begin{aligned}
\text{Manager} &\sqsubseteq \text{Employee}\\
\text{AreaManager} &\sqsubseteq \text{Manager}\\
\text{TopManager} &\sqsubseteq \text{Manager}\\
\text{AreaManager} &\sqsubseteq \neg\text{TopManager}
\end{aligned}
$$

$$
\begin{aligned}
\text{Employee} &\sqsubseteq \exists \mathit{salary}\\
\exists \mathit{salary}^- &\sqsubseteq \texttt{xsd:int}\\
&(\textbf{funct}\ \mathit{salary})
\end{aligned}
$$

$$
\begin{aligned}
\exists\text{worksFor} &\sqsubseteq \text{Employee}\\
\exists\text{worksFor}^- &\sqsubseteq \text{Project}\\
\text{Employee} &\sqsubseteq \exists\text{worksFor}\\
\text{Project} &\sqsubseteq \exists\text{worksFor}^-
\end{aligned}
$$

$$
\begin{aligned}
\exists\text{manages} &\sqsubseteq \text{TopManager}\\
\exists\text{manages}^- &\sqsubseteq \text{Project}\\
\text{TopManager} &\sqsubseteq \exists\text{manages}\\
\text{Project} &\sqsubseteq \exists\text{manages}^-\\
\text{manages} &\sqsubseteq \text{worksFor}\\
&(\textbf{funct}\ \text{manages})\\
&(\textbf{funct}\ \text{manages}^-)\\
&\cdots
\end{aligned}
$$

*Note:* *DL-Lite* cannot capture completeness of a hierarchy. This would require **disjunction** (i.e., **OR**).

# Query answering in *DL-Lite*

Based on query rewriting: given a (U)CQ $q$ and an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$:

1. **Compute the perfect rewriting of $q$ w.r.t. $\mathcal{T}$**, which is a FOL query.
2. **Evaluate the perfect rewriting** over $\mathcal{A}$. (Ignore the mapping for now.)

To compute the perfect rewriting, starting from $q$, iteratively get a CQ to be processed and either:

- **expand an atom** using an inclusion axiom, or
- **unify** atoms in the CQ to obtain a more specific CQ to be further expanded.

Each result of the above steps is added to the queries to be processed.

We can restrict expansion and unification so as to ensure termination without losing completeness.

*Note:* negative inclusions and functionalities play a role in ontology satisfiability, but can be ignored during query answering (i.e., we have **separability**).

# Query answering in *DL-Lite* – Example

TBox:

$\mathsf{Coordinator} \sqsubseteq \mathsf{PrincipalInv}$ $\quad \forall x(\mathsf{Coordinator}(x) \rightarrow \mathsf{PrincipalInv}(x))$

$\mathsf{PrincipalInv} \sqsubseteq \exists \mathsf{worksFor}$ $\quad \forall x(\mathsf{PrincipalInv}(x) \rightarrow \exists y(\mathsf{worksFor}(x, y)))$

$\exists \mathsf{worksFor}^- \sqsubseteq \mathsf{Project}$ $\quad \forall x(\exists y(\mathsf{worksFor}(y, x)) \rightarrow \mathsf{Project}(x))$

Query: $\mathsf{q}(x) \leftarrow \mathsf{worksFor}(x, y), \mathsf{Project}(y)$

Perfect rewriting:

$\mathsf{q}(x) \leftarrow \mathsf{worksFor}(x, y), \mathsf{Project}(y)$

$\mathsf{q}(x) \leftarrow \mathsf{worksFor}(x, y), \mathsf{worksFor}(\_, y)$

$\mathsf{q}(x) \leftarrow \mathsf{worksFor}(x, \_)$

$\mathsf{q}(x) \leftarrow \mathsf{PrincipalInv}(x)$

$\mathsf{q}(x) \leftarrow \mathsf{Coordinator}(x)$

ABox: $\mathsf{worksFor}(\mathtt{arild}, \mathtt{optique})$ $\quad \mathsf{Coordinator}(\mathtt{arild})$

$\mathsf{worksFor}(\mathtt{ian}, \mathtt{seals})$ $\quad \mathsf{Coordinator}(\mathtt{franz})$

Evaluating the queries over the ABox (seen as a DB) produces as answer $\{\mathtt{arild}, \mathtt{ian}, \mathtt{franz}\}$.

Optique unibz.it

# Complexity of reasoning in *DL-Lite*

Ontology satisfiability and all classical DL reasoning tasks are:

- Efficiently tractable in the size of the TBox (i.e., PTIME).
- Very efficiently tractable in the size of the ABox (i.e., $AC^0$).

In fact, reasoning can be done by constructing suitable FOL/SQL queries and evaluating them over the ABox (FOL-rewritability).

Query answering for CQs and UCQs is:

- PTIME in the size of the TBox.
- $AC^0$ in the size of the ABox.
- Exponential in the size of the **query** (NP-complete).
  Bad? . . . not really, this is exactly as in relational DBs.

So, is everything solved and can we deploy this technology?
**NO! Efficiency is a major issue!**

# Outline

# Size of the rewriting

Even in the presence of a shallow hierarchy in the TBox, the perfect rewriting, when expressed as a UCQ, may be very large in the original query.

Consider the query: $q(x) \leftarrow A_1(x), A_2(x), \ldots, A_n(x)$

and the concept inclusions: $B_1 \sqsubseteq A_1 \quad \cdots \quad B_n \sqsubseteq A_n$.

In the UCQ perfect rewriting of $q$, we have all CQs of the form

$$q(x) \leftarrow Z_1(x), Z_2(x), \ldots, Z_n(x)$$

where each $Z_i$ stands for either $A_i$ or $B_i$.

**There are $2^n$ such CQs!**
Unfortunately, this blowup **occurs also in practice**.

# Optimizing OBDA

- Novel techniques ensure more compact rewritings [Kontchakov *et al.*, 2010; Rosati and Almatelli, 2010; Gottlob and Schwentick, 2012; Kikot *et al.*, 2012].
- However, what matters is the **efficiency of evaluation by the DBMS**, not the size of the rewriting.
- Optimizing rewriting is necessary but not sufficient, since the more compact rewritings are much more difficult to evaluate.

**Holistic approach** is required, that considers all components of an OBDA system, i.e.:

- the TBox $\mathcal{T}$,
- the mappings $\mathcal{M}$, and
- the data sources $\mathcal{S}$ with their constraints.

# Exploiting data constraints for optimization

- Ontology TBox pruning using constraints and mappings.
- Semantic index to efficiently deal with concept hierarchies.
- Rewriting optimization using constraints and mappings.

The constraints on the data induce via the mapping $\mathcal{M}$ between $\mathcal{T}$ and $\mathcal{S}$ **dependencies in the ABox**.

# ABox dependencies

ABox dependencies express **constraints on the data in the ABox**.

- Syntax: $C_1 \preceq C_2$ $R_1 \preceq R_2$
- Meaning: constrain the assertions present in the ABox
  - $\mathcal{A} \vdash A_1 \preceq A_2$ iff $A_1(d) \in \mathcal{A}$ implies $A_2(d) \in \mathcal{A}$
  - $\mathcal{A} \vdash A \preceq \exists P$ iff $A(d) \in \mathcal{A}$ implies $P(d, d') \in \mathcal{A}$ for some $d'$
  - $\mathcal{A} \vdash P_1 \preceq P_2$ iff $P_1(d, d') \in \mathcal{A}$ implies $P_2(d, d') \in \mathcal{A}$
  - $\cdots$

*Note:* ABox dependencies are fundamentally different from TBox assertions. They constrain the syntactic level (the ABox itself), and not the models.

# Avoiding redundant inferences

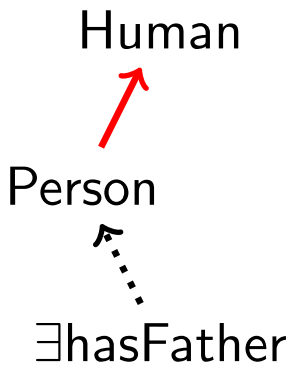
ABox dependencies can be exploited to avoid redundant inferences in query answering:

- Optimize the TBox by eliminating TBox assertions while maintaining completeness of query answering wrt ABoxes satisfying the dependencies.
- Optimize query rewriting algorithm so as to eliminate rewritings that are redundant wrt query containment under dependencies.

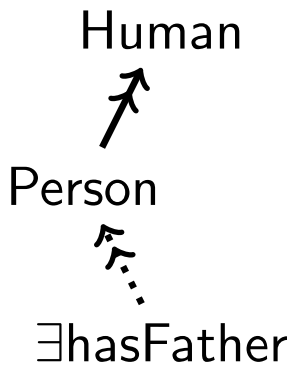
# Characterizing redundancy – Direct redundancy case 1

TBox optimization is based on a characterization of TBox assertions that are **redundant** wrt a set of ABox dependencies.

Let $\mathcal{T}$ be the hierarchy:




'$\rightarrow$' is redundant in $\mathcal{T}$ if $\Sigma$ is:

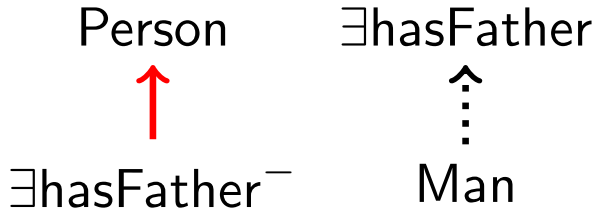



Note: $\Sigma$ may enforce e.g., that

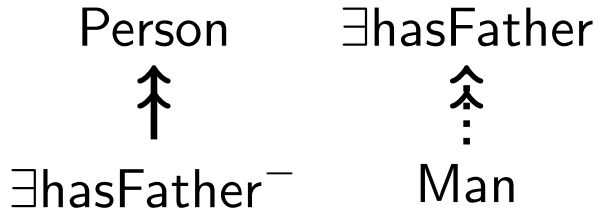
$$\text{hasFather}(\text{luisa}, \text{franz}) \in \mathcal{A} \quad \text{implies} \quad \text{Human}(\text{luisa}) \in \mathcal{A}.$$

# Characterizing redundancy – Direct redundancy case 2

Let $\mathcal{T}$ be the following TBox:




'$\rightarrow$' is redundant if $\Sigma$ is:




Note: $\Sigma$ may enforce e.g., that

$\text{Man}(\text{john}) \in \mathcal{A}$ implies there is some $d$ s.t. $\text{hasFather}(\text{john}, d) \in \mathcal{A}$ and $\text{Person}(d) \in \mathcal{A}$.

Note: '$\rightarrow$' would **not** be redundant if in $\Sigma$ we have:

$$\text{Man} \not\sqsubseteq \exists \text{hasFather}.$$

# TBox optimization

A TBox assertion might also be indirectly redundant (more complicated definition).

**Optimized query answering** wrt a TBox $\mathcal{T}$ and a set of ABox dependencies $\Sigma$:

1. Compute the deductive closure $\mathcal{T}_{cl}$ of $\mathcal{T}$ (at most quadratic in size of $\mathcal{T}$).
2. Compute the deductive closure $\Sigma_{cl}$ of $\Sigma$ (at most quadratic in size of $\Sigma$).
3. Eliminate from $\mathcal{T}_{cl}$ all TBox assertions that are redundant wrt $\Sigma_{cl}$, obtaining a TBox $\mathcal{T}_{opt}$.
4. Rewrite the query wrt $\mathcal{T}_{opt}$.
5. Evaluate the rewritten query over the ABox.

*Notes:*

- $\mathcal{T}_{opt}$ can be computed in polynomial time in the size of $\mathcal{T}$ and $\Sigma$.
- $\mathcal{T}_{opt}$ might be much smaller than $\mathcal{T}$.
- $\mathcal{T}_{opt}$ can be used instead of $\mathcal{T}$ independently of the adopted query rewriting method (provided the ABox satisfies $\Sigma$).

Optique unibz.it

# References I

[Berardi *et al.*, 2005] Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo.

Reasoning on UML class diagrams.

*Artificial Intelligence*, 168(1–2):70–118, 2005.

[Calvanese *et al.*, 1998] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini.

On the decidability of query containment under constraints.

In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[Calvanese *et al.*, 2004] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati, and Guido Vetere.

*DL-Lite*: Practical reasoning for rich DLs.

In *Proc. of the 17th Int. Workshop on Description Logic (DL 2004)*, volume 104 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, 2004.

[Calvanese *et al.*, 2005a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati.

Tailoring OWL for data intensive ontologies.

In *Proc. of the 1st Int. Workshop on OWL: Experiences and Directions (OWLED 2005)*, volume 188 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, 2005.

Optique unibz.it

# References II

[Calvanese *et al.*, 2005b] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati.

*DL-Lite*: Tractable description logics for ontologies.

In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

[Calvanese *et al.*, 2006] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati.

Data complexity of query answering in description logics.

In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, 2006.

[Calvanese *et al.*, 2007a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati.

Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family.

*J. of Automated Reasoning*, 39(3):385–429, 2007.

# References III

[Calvanese *et al.*, 2007b] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati.

Actions and programs over description logic ontologies.

In *Proc. of the 20th Int. Workshop on Description Logic (DL 2007)*, volume 250 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, pages 29–40, 2007.

[Calvanese *et al.*, 2007c] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz.

Answering regular path queries in expressive description logics: An automata-theoretic approach.

In *Proc. of the 22nd AAAI Conf. on Artificial Intelligence (AAAI 2007)*, pages 391–396, 2007.

[Calvanese *et al.*, 2008a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Riccardo Rosati, and Marco Ruzzi.

Data integration through *DL-Lite*$_{\mathcal{A}}$ ontologies.

In Klaus-Dieter Schewe and Bernhard Thalheim, editors, *Revised Selected Papers of the 3rd Int. Workshop on Semantics in Data and Knowledge Bases (SDKB 2008)*, volume 4925 of *Lecture Notes in Computer Science*, pages 26–47. Springer, 2008.

Optique unibz.it

# References IV

[Calvanese *et al.*, 2008b] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati.

Path-based identification constraints in description logics.

In *Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 231–241, 2008.

[Calvanese *et al.*, 2008c] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini.

Conjunctive query containment and answering under description logics constraints.

*ACM Trans. on Computational Logic*, 9(3):22.1–22.31, 2008.

[Eiter *et al.*, 2008] Thomas Eiter, Georg Gottlob, Magdalena Ortiz, and Mantas Šimkus.

Query answering in the description logic Horn-$\mathcal{SHIQ}$.

In *Proc. of the 11th Eur. Conference on Logics in Artificial Intelligence (JELIA 2008)*, pages 166–179, 2008.

[Eiter *et al.*, 2009] Thomas Eiter, Carsten Lutz, Magdalena Ortiz, and Mantas Šimkus.

Query answering in description logics with transitive roles.

In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, pages 759–764, 2009.

# References V

[Glimm *et al.*, 2008a] Birte Glimm, Ian Horrocks, Carsten Lutz, and Uli Sattler.

Conjunctive query answering for the description logic $\mathcal{SHIQ}$.

*J. of Artificial Intelligence Research*, 31:151–198, 2008.

[Glimm *et al.*, 2008b] Birte Glimm, Ian Horrocks, and Ulrike Sattler.

Unions of conjunctive queries in $\mathcal{SHOQ}$.

In *Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 252–262, 2008.

[Gottlob and Schwentick, 2012] Georg Gottlob and Thomas Schwentick.

Rewriting ontological queries into small nonrecursive Datalog programs.

In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012)*, 2012.

[Kikot *et al.*, 2012] Stanislav Kikot, Roman Kontchakov, and Michael Zakharyaschev.

Conjunctive query answering with OWL 2 QL.

In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 275–285, 2012.

Optique unibz.it

# References VI

[Kontchakov *et al.*, 2010] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev.

The combined approach to query answering in *DL-Lite*.

In *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 247–257, 2010.

[Levy and Rousset, 1998] Alon Y. Levy and Marie-Christine Rousset.

Combining Horn rules and description logics in CARIN.

*Artificial Intelligence*, 104(1–2):165–209, 1998.

[Lutz, 2008] Carsten Lutz.

The complexity of conjunctive query answering in expressive description logics.

In *Proc. of the 4th Int. Joint Conf. on Automated Reasoning (IJCAR 2008)*, volume 5195 of *Lecture Notes in Artificial Intelligence*, pages 179–193. Springer, 2008.

[Ortiz *et al.*, 2006] Maria Magdalena Ortiz, Diego Calvanese, and Thomas Eiter.

Characterizing data complexity for conjunctive query answering in expressive description logics.

In *Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006)*, pages 275–280, 2006.

# References VII

[Ortiz *et al.*, 2008] Magdalena Ortiz, Diego Calvanese, and Thomas Eiter.

Data complexity of query answering in expressive description logics via tableaux.

*J. of Automated Reasoning*, 41(1):61–98, 2008.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati.

Linking data to ontologies.

*J. on Data Semantics*, X:133–173, 2008.

[Rodríguez-Muro and Calvanese, 2011a] Mariano Rodríguez-Muro and Diego Calvanese.

Dependencies to optimize ontology based data access.

In *Proc. of the 24th Int. Workshop on Description Logic (DL 2011)*, volume 745 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, 2011.

[Rodriguez-Muro and Calvanese, 2011b] Mariano Rodriguez-Muro and Diego Calvanese.

Semantic index: Scalable query answering without forward chaining or exponential rewritings.

In *Posters of the 10th Int. Semantic Web Conf. (ISWC 2011)*, 2011.

Optique unibz.it

# References VIII

[Rodriguez-Muro and Calvanese, 2012] Mariano Rodriguez-Muro and Diego Calvanese.

High performance query answering over *DL-Lite* ontologies.

In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 308–318, 2012.

[Rodríguez-Muro, 2010] Mariano Rodríguez-Muro.

*Tools and Techniques for Ontology Based Data Access in Lightweight Description Logics*.

PhD thesis, KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, 2010.

[Rosati and Almatelli, 2010] Riccardo Rosati and Alessandro Almatelli.

Improving query answering over *DL-Lite* ontologies.

In *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 290–300, 2010.