Data Modeling and AI: from Semantic Networks to Knowledge Graphs

Maurizio Lenzerini

lenzerini@diag.uniroma1.it
Dipartimento di Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti
Sapienza Università di Roma, Italy

12th International Conference on Data Science, Technology and Applications
DATA 2023
Roma (Italy), July 11, 2023

Data Modeling and Artificial Intelligence

- Data Modeling
  The discipline aiming at defining how data are structured at various levels:
  - conceptual (characterizes the domain of interest that data describe)
  - logical (characterizes the data organization seen by the user of the Data Manager)
  - physical (characterize how data are physically stored in the appropriate medium managed by the Data Manager)

- Artificial Intelligence
  The discipline aiming at understanding, representing, and simulating various forms of human intelligence by machines.
  - Knowledge-driven (symbolic AI) (the world is specified by a symbolic structure properly designed)
    $\longrightarrow$ *Knowledge Representation*
  - Data-driven (connectionist AI) (the world emerges from the experience coded into data)
    $\longrightarrow$ *Machine Learning*

SAPIENZA
UNIVERSITÀ DI ROMA

Knowledge Representation and Reasoning

Knowledge representation and reasoning (KR and KRR) is the area of Artificial Intelligence (AI) concerned with how knowledge can be represented symbolically and manipulated in an automated way by reasoning programs. More informally, it is the part of AI that is concerned with domain modeling, thinking about such model and taking adavantages of reasoning to contribute to intelligent behavior. [Brachman & Levesque, 2004]

Structured Knowledge Representation languages, based on:

- individual objects
- concepts
- relations between concepts
- rules expressing properties of the above elements

Let us start with a short summary of the hystory of a long journey

Note:

- Semantic Network, Frame Systems: specification of concepts and relationships that are relevant in the domain of interest
- Data model (or Data Schema): specification of data items that are relevant in a certain context

The early days ... (a semantic network in the middle age)

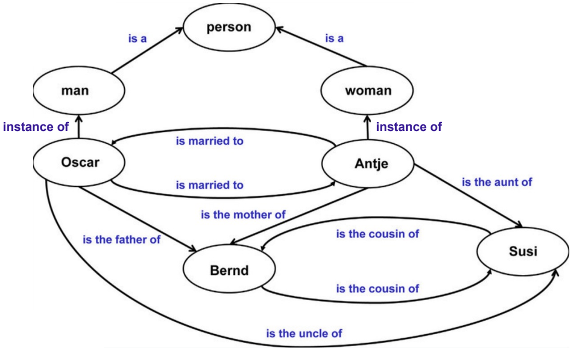
The beginning ('50, '60s and '70s)

- '50s: "Semantic Nets" (R. H. Richens 1956), an "interlingua" for machine translation
- '60s: Network data model (IDS) and Hierchical data model (IMS) no modeling or design theory
- '60s: Semantic Networks (R. Quillian, 1968)
- '70s: Relational data model (E. Codd, 1970) normalization theory
- '70s: Frame-based systems (M. Minsky, 1974)
- '70s: Database modeling (Roussolopoulos & J. Mylopoulos, 1975)
- '70s: "What's in a link" (W.A. Woods, 1975)
- '70s: Entity-Relationship model (P. Chen, 1976)
- '70s: "Conceptual Graphs" (J. Sowa, 1976)
- '70s: "Aggregation and Generalization" (Smith & Smith, 1977)
- '70s: "KL-ONE" (R. Brachman et al, 1977) automated reasoning
- '70s: "Data and reality" (W. Kent, 1978)

Evolution ('80s, '90s and '00s)

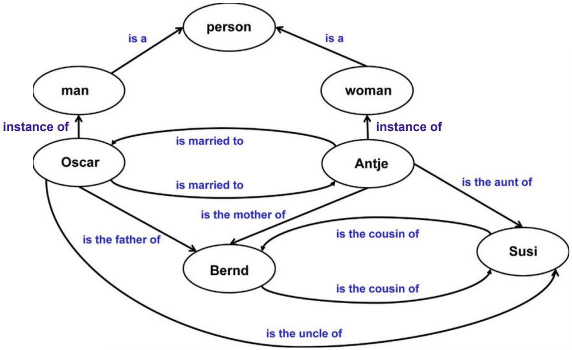
- '80s: Design methods based on normalization (C. Date, 1981)
- '80s: "Towards a Logical Reconstruction of Relational Database Theory" (R. Reiter 1982)
- '80s: "Conceptual modeling" (Broadie et al 1984)
- '80s: "Expressiveness and tractability in knowledge representation and reasoning" (H. Levesque & Brachman 1987)
- '90s: Unified Modeling Language (Fowler & Scott 1997)
- '90s: Description Logics
- '90s: Semistructured data models and NoSQL
- '00s: OWL (Ontology Web Language)
- '00s: Graph databases
- '00s: Google announces its knowledge graph (2012)

Semantic networks and knowledge graphs

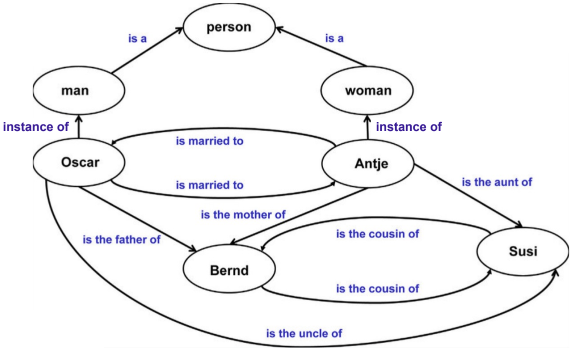


Semantic Network (1968)

Semantic networks and knowledge graphs



Semantic Network (1968)



Knowledge Graph (2022)

Really no differences?

SAPIENZA UNIVERSITÀ DI ROMA

A lot of differences in ...

1. Formalization
2. Basic deductive reasoning
3. Mapping data to knowledge
4. Query answering
5. Other types of reasoning

Outline of the talk

Logic-based formalization

In 1977, Brachman, Woods, and others started developing a precursor of current KR systems: KL-ONE [Woods & Brachman 1977].

Use logic ...

... specifically designed to represent class-oriented structured knowledge:

Consider the domain as composed of objects (constants), organized into:

- Concepts, denoting sets of objects (unary predicates)
- Roles, denoting binary relations on objects (binary predicates)

Knowledge on the domain asserted through statements (axioms)

80's: Brachman & Levesque – AAAI 1984

In KL-ONE (1977), axioms belong to two components:

- Terminological component (TBox), expressing intensional knowledge, e.g., $C \sqsubseteq D$ (i.e. C is-a D), where C,D are (structured) concepts
- Assertional component (ABox), with extensional assertions, e.g., C(a) or R(a,b), where a,b are individuals and R is a relationship (also called role)

Ron Brachman & Hector Levesque – The Tractability of Subsumption in Frame-Based Description Languages [Brachman & Levesque 1984, AAAI]:

- Use **Description Logics** and focus on relevant deduction tasks (e.g., concept subsumption)
- Use **complexity** of deduction tasks to understand the intrinsic computational properties of reasoning in the description logic
- Study the **tradeoff** between expressivity and complexity and find **effective/tractable** logics

These aspects will be crucial in most of the research carried out in KR in general and in Description Logics in particular

Sapienza Università di Roma

Outline of the talk

Reasoning about concept descriptions

The problem addressed in [Brachman & Levesque 1984, AAAI] stems from the need of reasoning about frame expressions

Frame expression C: A person that has at least one child, all of whose sons are lawyers and all of whose daughters are doctors

[person

child	:	(≥ 1)
son	:	lawyer
daughter	:	doctor]

formalized into logic-based concept descriptions, e.g.,

Concept description for C

$$\text{person} \sqcap (\exists \text{child}) \sqcap (\forall \text{son.lawyer}) \sqcap (\forall \text{daughter.doctor})$$

Reasoning task (classification): Build a *is-a* hierarchy with relevant concept descriptions in such a way that (C_1 *is-a* C_2) iff C_2 subsumes C_1, i.e., $C_1 \sqsubseteq C_2$.

SAPIENZA
UNIVERSITÀ DI ROMA

Two basic concept description languages studied in [Brachman & Levesque 1984, AAAI]

$\mathcal{FL}$:

$$\begin{array}{rcl} C, D & \longrightarrow & A \mid C \sqcap D \mid \forall R.C \mid \exists R \\ R & \longrightarrow & P \mid \ (\textsc{restr}\ R\ C) \end{array}$$

$\mathcal{FL}^-$:

$$\begin{array}{rcl} C, D & \longrightarrow & A \mid C \sqcap D \mid \forall R.C \mid \exists R \\ R & \longrightarrow & P \end{array}$$

Semantics (given a first-order interpretation $\mathcal{I}$):

$$\begin{array}{rcl} (C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ (\forall R.C)^{\mathcal{I}} & = & \{a : \forall b.\, (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\ (\exists R)^{\mathcal{I}} & = & \{a : \exists b.\, (a,b) \in R^{\mathcal{I}}\} \\ (\textsc{restr}\ \ R\ C)^{\mathcal{I}} & = & \{(a,b) \in R^{\mathcal{I}} : b \in C^{\mathcal{I}}\} \end{array}$$

Example of description in $\mathcal{FL}$:

person ⊓ (∃child) ⊓ (∀daughter.doctor) ⊓ ∀(RESTR son lawyer).rich

Sapienza Università di Roma

Checking subsumption in $\mathcal{FL}^-$ (structural algorithm)

Let $C = C_1 \sqcap \cdots \sqcap C_n$ and $D = D_1 \sqcap \cdots \sqcap D_m$ be in normal form (obtained by pushing $\sqcap$). We want to check whether C subsumes D, i.e., $D \sqsubseteq C$.

Algorithm $\texttt{SUBS}(C, D)$:
Return $true$ if and only if for all C_i:

1. if C_i is either an atomic concept, or is of the form $\exists R$, then there exists j such that D_j is exactly C_i;
2. if C_i is of the form $\forall R.C'$, then there exists j such that D_j is of the form $\forall R.D'$ (same atomic role R) with $\texttt{SUBS}(C', D') = true$.

Theorem in [Brachman & Levesque 1984, AAAI]

$\texttt{SUBS}(C, D)$ terminates, returning $true$ if and only if $(D \sqsubseteq C)$. Moreover, its complexity is $O(|C| \times |D|)$ and therefore it runs in polynomial time.

Bell Labs developed the CLASSIC system with tractable subsumption
[Borgida, Brachman, McGuinness & Resnick 1989, SIGMOD], [Patel-Schneider, McGuinness, Brachman, Resnick & Borgida 1991, SIGART]

Checking subsumption in $\mathcal{FL}$

Notice that:

$$\forall(\text{RESTR R C}).Z \sqcap \forall(\text{RESTR R D}).Z$$
$$\equiv$$
$$\forall(\text{RESTR } R\ (C \sqcup D)).Z$$

and that $\forall(\text{RESTR } R\ (C \sqcup D)).Z$ is subsumed by $\forall R.Z$ if and only if $(C \sqcup D) \equiv \top$.

Theorem in [Brachman & Levesque 1984, AAAI]

Checking subsumption in $\mathcal{FL}$ is coNP-hard.

Later, the problem has been shown to be PSPACE-complete [Donini & al 1997, InfComp].

Checking subsumption in $\mathcal{FL}^+$

$\mathcal{FL}^+$:

$$C, D \longrightarrow A \mid C \sqcap D \mid \forall R.C \mid \exists R \mid R \subseteq Q$$
$$R \longrightarrow P \mid (\textsc{restr}\ R\ C) \mid R \circ Q$$

Semantics (given a first-order interpretation $\mathcal{I}$):

$$(R \subseteq Q)^{\mathcal{I}} = \{a : \forall b.\, (a,b) \in R^{\mathcal{I}} \rightarrow (a,b) \in Q^{\mathcal{I}}\}$$
$$(R \circ Q)^{\mathcal{I}} = \{(a,b) : \exists y.\ R(a,y) \wedge Q(y,b)\}$$

Male persons all of whose friends of friends are also his friends

$$\text{person} \sqcap \text{male} \sqcap (\text{friend} \circ \text{friend} \subseteq \text{friend})$$

Theorem in [Schmidt-Schauss 1989, KR]

Checking subsumption in $\mathcal{FL}^+$ is undecidable. The same holds for the description language of KL-ONE.

From expressions to TBox axioms (ontology)

- Subsumption check: $\models C \sqsubseteq D?$
- Reasoning with a set of TBox axioms $\mathcal{T}$ (called an **ontology**):

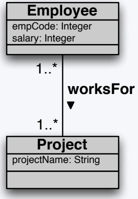
$$\mathcal{T} \models C \sqsubseteq D?$$

To reach the expressive power needed in real world scenarios (e.g., to capture conceptual models), we need:

- new constructs (e.g., inverse roles $\mathcal{I}$, qualified cardinality restrictions $\mathcal{Q}$, and more),
- TBox axioms

Ontology $\mathcal{O}$ (TBox)

Employee $\sqsubseteq \exists$worksFor
Employee $\sqsubseteq \exists$empCode
Employee $\sqsubseteq \exists$salary
Project $\sqsubseteq \exists$worksFor$^-$
Project $\sqsubseteq \exists$projectName
$\exists$worksFor $\sqsubseteq$ Employee
$\exists$worksFor$^-$ $\sqsubseteq$ Project



Reasoning under TBox axioms

- **Tableaux** for the expressive DL $\mathcal{ALC}$ with assertions [Schmidt-Schauss & Smolka 1991, AIJ], [Baader 1991, IJCAI], [Nebel 1991], [Donini, Lenzerini, Nardi & Nutt 1991, KR]

- **Description logics = Modal Logics for actions** [Schild 1991, IJCAI], [De Giacomo 1995, PhD]

  The Description Logic $\mathcal{C}$:

$$\begin{array}{lcl} C & \longrightarrow & \top \mid A \mid C_1 \sqcap C_2 \mid \neg C \mid \exists R.C \mid \forall R.C \\ R & \longrightarrow & P \mid R_1 \sqcup R_2 \mid R_1 \circ R_2 \mid R^* \mid id(C) \end{array}$$

  PDL:

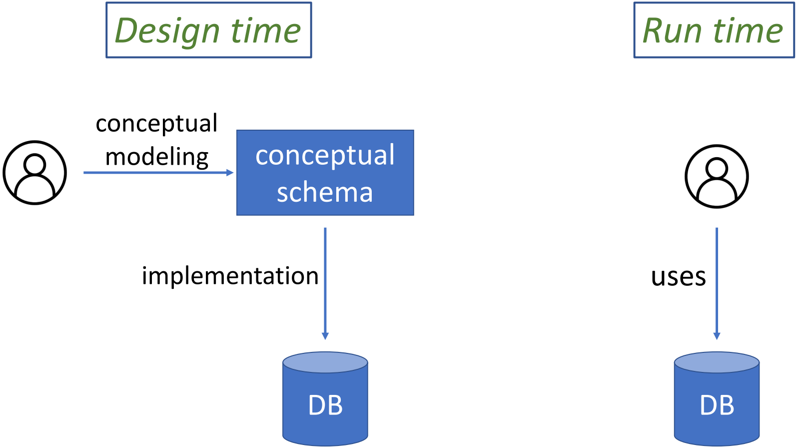
$$\begin{array}{lcl} \phi & \longrightarrow & true \mid A \mid \phi_1 \wedge \phi_2 \mid \neg \phi_1 \mid \langle r \rangle \phi_1 \mid [r] \phi_1 \\ r & \longrightarrow & P \mid r_1 \cup r_2 \mid r_1 ; r_2 \mid r^* \mid \phi? \end{array}$$

- **Optimized fast tableaux** for expressive DLs as $\mathcal{ALCQI}$, later $\mathcal{SHIQ}$ [Horrocks 1998, KR], [Horrocks, Sattler & Tobies 2000, IGPL], crucial for
  - the development of **reasoners**, such as FACT++, PELLET, RACER, and many others
  - the role played by the DL community on the ***OWL W3C Standard***
  - reasoning on conceptual models, e.g., UML [Berardi, Calvanese & De Giacomo 2005, AIJ], and developing information models in several domain (health, biology, finance, PA, Enterprise modeling, SE, IR, ...)

Outline of the talk

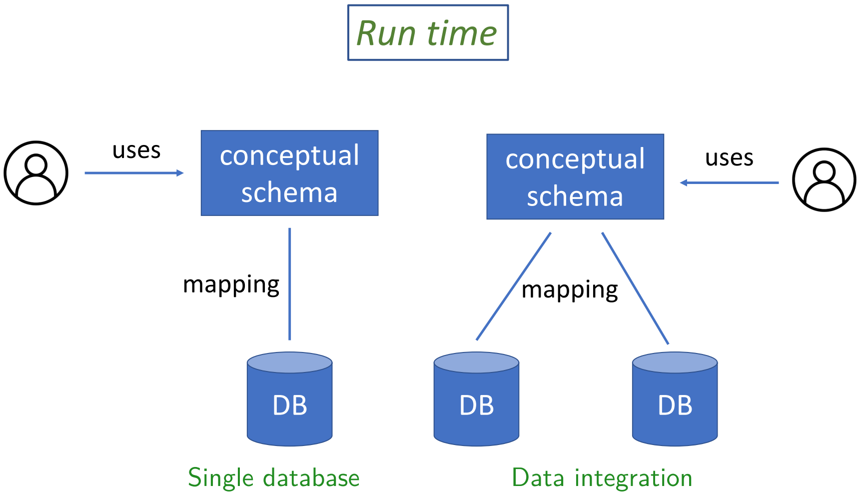
1. Formalization
2. Basic deductive reasoning
3. **Mapping data to knowledge**
4. Query answering
5. Other types of reasoning

Conceptual data modeling in the '80s: top-down data design



The conceptual schema disappears at run time!

Basic contribution of Knowledge Representation



Principles and tools for "using" the conceptual schema at run time in different scenarios

The need of abstract data representation

Fragment of a relational table in a Bank Information system:

CUC	TS_START	TS_END	ID_GRUP	FLAG_CP	FLAG_CF	FATTURATO	FLAG_FATT	
124589	30-lug-2004	1-gen-9999	92736	S	N	195000,00	N	
140904	15-mag-2001	15-giu-2005	35060	N	N	230600,00	N	
124589	5-mag-2001	30-lug-2004	92736	N	S	195000,00	S	
-452901	13-mag-2001	27-lug-2004	92770	S	N	392000,00	N	
129008	10-mag-2001	1-gen-9999	62010	N	S	247000,00	S	
-472900	10-mag-2001	1-gen-9999	62010	S	N	0 00	N	
130976	7-mag-2001	9-lug-2003	75680					

SAPIENZA
UNIVERSITÀ DI ROMA

The need of abstract data representation

Negative value denotes a holding

CUC	TS_START	TS_END	ID_GRUP	FLAG_CP	FLAG_CF	FATTURATO	FLAG_FATT	
124589	30-lug-2004	1-gen-9999	92736	S	N	195000,00	N	
140904	15-mag-2001	15-giu-2005	35060	N	N	230600,00	N	
124589	5-mag-2001	30-lug-2004	92736	N	S	195000,00	S	
-452901	13-mag-2001	27-lug-2004	92770	S	N	392000,00	N	
129008	10-mag-2001	1-gen-9999	62010	N	S	247000,00	S	
-472900	10-mag-2001	1-gen-9999	62010	S	N	0 00	N	
130976	7-mag-2001	9-lug-2003	75680					

SAPIENZA UNIVERSITÀ DI ROMA

The need of abstract data representation

S means that the customer is the leader of the group it belongs to

S means that the customer is the head of the group it belongs to

CUC	TS_START	TS_END	ID_GRUP	FLAG_CP	FLAG_CF	FATTURATO	FLAG_FATT
124589	30-lug-2004	1-gen-9999	92736	S	N	195000,00	N
140904	15-mag-2001	15-giu-2005	35060	N	N	230600,00	N
124589	5-mag-2001	30-lug-2004	92736	N	S	195000,00	S
-452901	13-mag-2001	27-lug-2004	92770	S	N	392000,00	N
129008	10-mag-2001	1-gen-9999	62010	N	S	247000,00	S
-472900	10-mag-2001	1-gen-9999	62010	S	N	0 00	N
130976	7-mag-2001	9-lug-2003	75680				

SAPIENZA UNIVERSITÀ DI ROMA

The need of abstract data representation

N means that the FATTURATO field is not valid

CUC	TS_START	TS_END	ID_GRUP	FLAG_CP	FLAG_CF	FATTURATO	FLAG_FATT
124589	30-lug-2004	1-gen-9999	92736	S	N	195000,00	N
140904	15-mag-2001	15-giu-2005	35060	N	N	230600,00	N
124589	5-mag-2001	30-lug-2004	92736	N	S	195000,00	S
-452901	13-mag-2001	27-lug-2004	92770	S	N	392000,00	N
129008	10-mag-2001	1-gen-9999	62010	N	S	247000,00	S
-472900	10-mag-2001	1-gen-9999	62010	S	N	0 00	N
130976	7-mag-2001	9-lug-2003	75680				

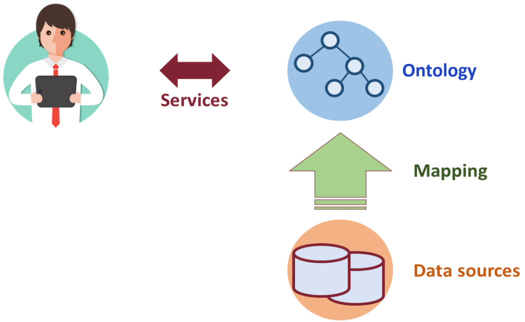
Managing data through the lens of an ontology: Ontology-based Data Management

The **Ontology-based Data Management** paradigm is rooted on the idea of using Database Theory fundamentals and Knowledge Representation and Reasoning techniques for a new way of managing data, and characterized by the following principles:

- Data may reside where they are (no need to move data)
- Build a conceptual specification of the domain of interest, in terms of knowledge structures
- Map such knowledge structures to concrete data sources
- Express all services over the knowledge structures
- Automatically translate knowledge services to data services

SAPIENZA UNIVERSITÀ DI ROMA

Ontology-based data management (OBDM)



Based on three main components $\langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$ [Poggi & al. 2008, InfSys]:

- Data sources, representing a set of heterogeneous data repositories
- Ontology, a declarative specification of the domain of interest, expressed as a TBox in a DL
- Mappings, used to semantically link data at the sources to the ontology

Potential benefits of managing data through the lens of an ontology: in making sense of data, in assessing data quality, in integrating, cleaning and exchanging data, etc.

OBDM: semantics

Let $\mathcal{J} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$ be an OBDM specification, and D an $\mathcal{S}$-database.

Def.: Semantics

The semantics of $(\mathcal{J}, D)$ is given by its models $mod(\mathcal{J}, D)$, where a model is an interpretation $\mathcal{I}$ for $\mathcal{O}$ such that:

- $\mathcal{I}$ satisfies all axioms of $\mathcal{O}$,
- $\mathcal{I}$ satisfies $\mathcal{M}$ wrt D, i.e., satisfies every assertion

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

  in $\mathcal{M}$ wrt D, which means that the sentence $\forall \vec{x}\ (\Phi(\vec{x}) \rightarrow \Psi(\vec{x}))$ is true in $\mathcal{I} \cup D$.

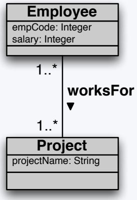
$(\mathcal{J}, D)$ is **consistent**, if it admits at least one model.

SAPIENZA UNIVERSITÀ DI ROMA

Ontology-based data management specification – Example

Ontology $\mathcal{O}$ (TBox)

Employee $\sqsubseteq$ $\exists$worksFor
Employee $\sqsubseteq$ $\exists$empCode
Employee $\sqsubseteq$ $\exists$salary
Project $\sqsubseteq$ $\exists$worksFor$^-$
Project $\sqsubseteq$ $\exists$projectName
$\exists$worksFor $\sqsubseteq$ Employee
$\exists$worksFor$^-$ $\sqsubseteq$ Project



Source schema $\mathcal{S}$

$D_1[\mathit{SSN} : \mathit{String}, \mathit{PrName} : \mathit{String}]$
Employees and Projects they work for

$D_2[\mathit{Code} : \mathit{String}, \mathit{Salary} : \mathit{Int}]$
Employee's Code with salary

$D_3[\mathit{Code} : \mathit{String}, \mathit{SSN} : \mathit{String}]$
Employee's Code with SSN

...

Mapping $\mathcal{M}$

M_1:
```
SELECT SSN,PrName
FROM D1
```
$\rightarrow$ Employee(**pers**(SSN)), Project(**proj**($PrName$)), projectName(**proj**($PrName$), $PrName$), workFor(**pers**(SSN), **proj**($PrName$))

M_2:
```
SELECT SSN,Salary
FROM D2, D3
WHERE D2.Code = D3.Code
```
$\rightarrow$ Employee(**pers**(SSN)), salary(**pers**(SSN), $Salary$)

Ontology-based data management: issues

- *Ontology-based data access (OBDA)* – also called Ontology-mediated query answering (OMQA)
- *Ontology-based data quality*
- *Ontology-based data cleaning*
- *Ontology-based data provenance*
- *Ontology-based data governance*
- *Ontology-based data restructuring*
- *Ontology-based business intelligence*
- *Ontology-based data exchange and coordination*
- *Ontology-based data abstraction*

Outline of the talk

Sapienza Università di Roma

From expressions to TBox axioms to queries

- Subsumption check:

$$\models C \sqsubseteq D?$$

- Reasoning with a set of TBox axioms $\mathcal{T}$:

$$\mathcal{T} \models C \sqsubseteq D?$$

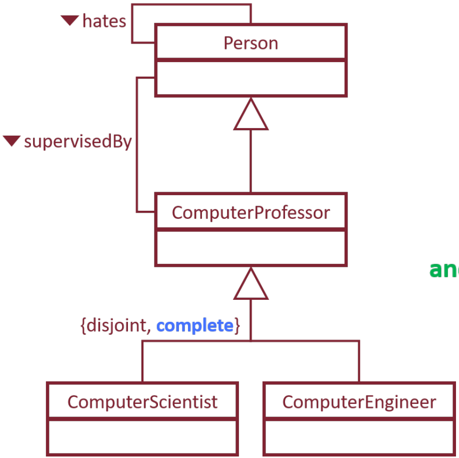
- Computing the **certain answers** to query $q(\vec{x})$ posed to $(\mathcal{J}, D)$ (and therefore over $\mathcal{O}$), i.e., those tuples $\vec{c}$ that satisfy q in **every model** in $mod(\mathcal{J}, D)$, i.e., such that

$$(\mathcal{J}, D) \models q(\vec{c})$$

$\Rightarrow$ **Answering conjunctive queries** on expressive DLs is decidable [Calvanese, De Giacomo & Lenzerini 1998, PODS]

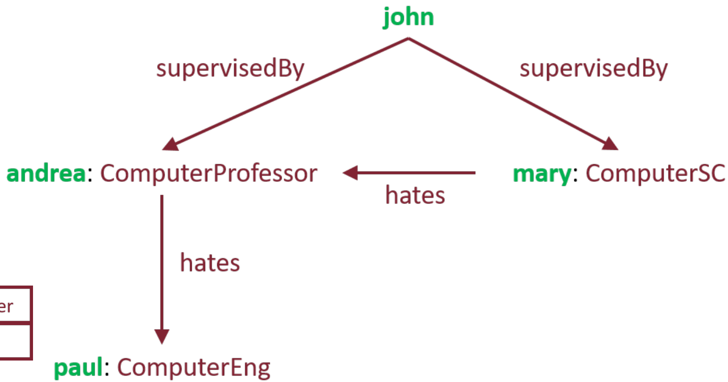
SAPIENZA UNIVERSITÀ DI ROMA

Computing certain answer – Example



Note that ComputerProfessor is partitioned into ComputerScientist and ComputerEngineer.

(Virtual) **knowledge graph** corresponding to $\mathcal{M}(D)$:

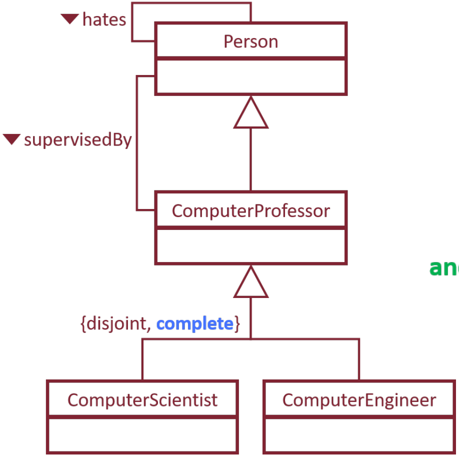


Query:

$\{ \ (x) \mid \exists y, z \, \text{supervisedBy}(x, y), \text{ComputerSC}(y), \text{hates}(y, z), \text{ComputerEng}(z) \ \}$

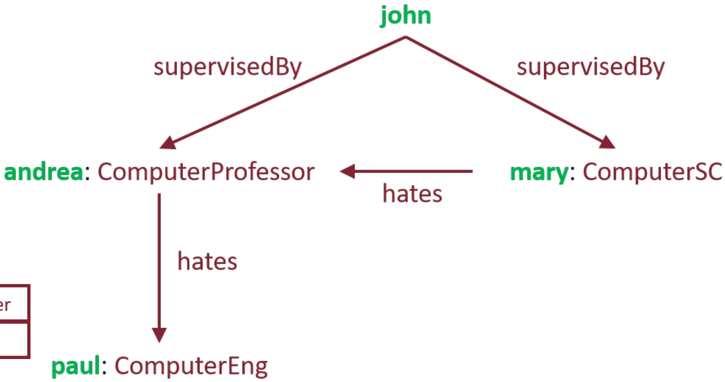
Answer: ???

Computing certain answer – Example



Note that ComputerProfessor is partitioned into ComputerScientist and ComputerEngineer.

(Virtual) **knowledge graph** corresponding to $\mathcal{M}(D)$:



Query:

$\{ \ (x) \mid \exists y, z \, \text{supervisedBy}(x, y), \text{ComputerSC}(y), \text{hates}(y, z), \text{ComputerEng}(z) \ \}$

Answer: { john } To obtain this answer, we need to **reason by cases** on the data

Sapienza Università di Roma

Complexity of conjunctive query answering in DLs

	Combined complexity	Data complexity
Plain databases	NP-complete	in LogSpace
OWL 2	2ExpTime-hard	coNP-hard [1]

[1] Already for a TBox with a single disjunction (see example above).

Complexity of conjunctive query answering in DLs

	Combined complexity	Data complexity
Plain databases	NP-complete	in LOGSPACE
OWL 2	2EXPTIME-hard	coNP-hard [(1)]

(1) Already for a TBox with a single disjunction (see example above).

Questions

- Can we find interesting DLs for which the query answering problem can be solved efficiently (i.e., in LOGSPACE wrt data complexity)?
- If yes, can we leverage relational database technology for query answering in OBDM?

SAPIENZA UNIVERSITÀ DI ROMA

Complexity of conjunctive query answering in DLs

	Combined complexity	Data complexity
Plain databases	NP-complete	in LogSpace
OWL 2	2ExpTime-hard	coNP-hard [1]

[1] Already for a TBox with a single disjunction (see example above).

Questions

- Can we find interesting DLs for which the query answering problem can be solved efficiently (i.e., in LogSpace wrt data complexity)?
- If yes, can we leverage relational database technology for query answering in OBDM?

$\Rightarrow$ **Lightweight DLs**, e.g., the *DL-Lite* [Calvanese & al 2005, 2007] and the $\mathcal{EL}$ [Baader & al 2005] families.

SAPIENZA UNIVERSITÀ DI ROMA

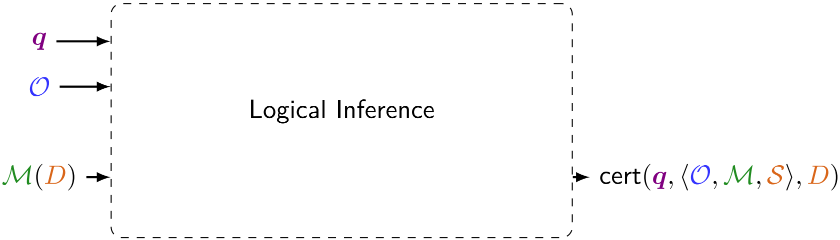
DL-Lite

DL-Lite TBox axioms have the form:

DL Syntax	FOL Syntax	Example	
$A_1 \sqsubseteq A_2$	$\forall x.A_1(x) \rightarrow A_2(x)$	Professor $\sqsubseteq$ Person	Positive Inclusions
$A \sqsubseteq \exists R$	$\forall x.A(x) \rightarrow \exists y.R(x,y)$	Professor $\sqsubseteq$ $\exists$teaches	
$\exists R \sqsubseteq A$	$\forall x,y.R(x,y) \rightarrow A(x)$	$\exists$teaches $\sqsubseteq$ Professor	
$\exists R^- \sqsubseteq A$	$\forall x,y.R(y,x) \rightarrow A(x)$	$\exists$teaches$^-$ $\sqsubseteq$ Student	
$\ldots$	$\ldots$	$\ldots$	
$A_1 \sqsubseteq \neg A_2$	$\forall x.A_1(x) \rightarrow \neg A_2(x)$	Professor $\sqsubseteq$ $\neg$Student	Negative Inclusions
$A \sqsubseteq \neg\exists R$	$\forall x.A(x) \rightarrow \neg\exists y.R(x,y)$	Student $\sqsubseteq$ $\neg\exists$teaches	
$\ldots$	$\ldots$	$\ldots$	
(**funct** R)	$\forall x,y,zR(x,y) \wedge R(x,z) \rightarrow y=z$	(**funct** hasAdvisor)	Functionalities
(**funct** R^-)	$\forall x,y,zR(y,x) \wedge R(z,x) \rightarrow y=z$	(**funct** isFatherOf$^-$)	

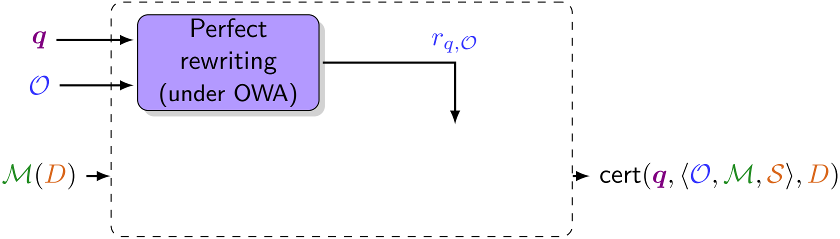
Crucial characteristic of *DL-Lite*: existence of a universal model

Query answering by query rewriting



To get "good" data complexity, the contribution of $\mathcal{M}$, D is separated from that of $\boldsymbol{q}$ and $\mathcal{O}$.

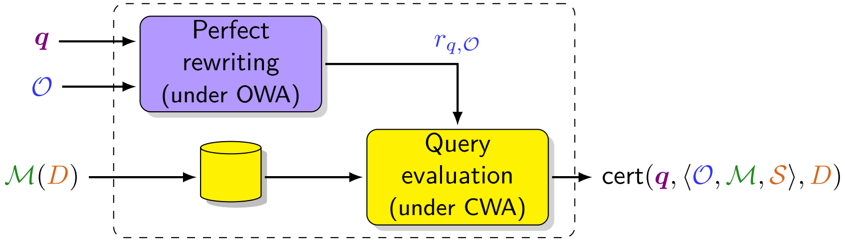
Query answering by query rewriting



To get "good" data complexity, the contribution of $\mathcal{M}$, D is separated from that of q and $\mathcal{O}$.

- $r_{q,\mathcal{O}}$ is a new query over $\mathcal{O}$, called the **perfect rewriting** of q w.r.t. $\mathcal{O}$

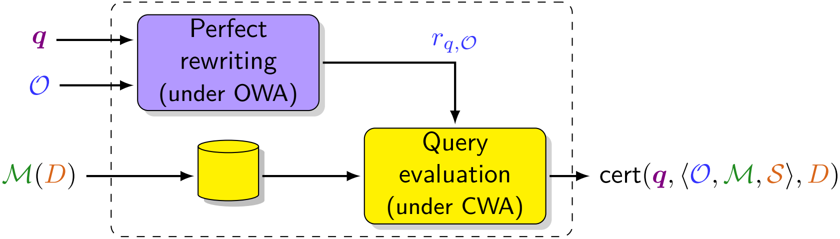
Query answering by query rewriting



To get "good" data complexity, the contribution of $\mathcal{M}$, D is separated from that of q and $\mathcal{O}$.

- $r_{q,\mathcal{O}}$ is a new query over $\mathcal{O}$, called the **perfect rewriting** of q w.r.t. $\mathcal{O}$
- evaluating $r_{q,\mathcal{O}}$ over $\mathcal{M}(D)$ can be done by evaluating $r_{q,\mathcal{O},\mathcal{M}}$ over D, where $r_{q,\mathcal{O},\mathcal{M}}$ is the **perfect rewriting** of q w.r.t. $\mathcal{O}$ and $\mathcal{M}$.

Query answering by query rewriting



To get "good" data complexity, the contribution of $\mathcal{M}$, D is separated from that of q and $\mathcal{O}$.

- $r_{q,\mathcal{O}}$ is a new query over $\mathcal{O}$, called the **perfect rewriting** of q w.r.t. $\mathcal{O}$
- evaluating $r_{q,\mathcal{O}}$ over $\mathcal{M}(D)$ can be done by evaluating $r_{q,\mathcal{O},\mathcal{M}}$ over D, where $r_{q,\mathcal{O},\mathcal{M}}$ is the **perfect rewriting** of q w.r.t. $\mathcal{O}$ and $\mathcal{M}$.

FO-rewritability of conjunctive query (CQ) answering

In *DL-Lite*, the perfect rewriting of a UCQ w.r.t. $\mathcal{O}$ and $\mathcal{M}$ is a **UCQ** whose size is polynomial in $\mathcal{O}$. Thus, **answering CQ in *DL-Lite* is in LogSpace in data complexity (i.e., w.r.t. D only)** and PTime in ontology complexity (i.e., w.r.t. $\mathcal{O}$ only).

Extensions to basic query answering

Extensions to basic query answering

- Bag semantics, aggregation operators, and counting [Nikolaou, Kostylev, Konstantinidis, Kaminski, Cuenca Grau & Horrocks 2019, AIJ], [Bienvenu, Manière & Thomazo 2020, IJCAI], [Calvanese, Corman, Lanti & Razniewski 2020, IJCAI]

SAPIENZA
UNIVERSITÀ DI ROMA

Extensions to basic query answering

- Bag semantics, aggregation operators, and counting [Nikolaou, Kostylev, Konstantinidis, Kaminski, Cuenca Grau & Horrocks 2019, AIJ], [Bienvenu, Manière & Thomazo 2020, IJCAI], [Calvanese, Corman, Lanti & Razniewski 2020, IJCAI]
- Explanation and provenance [Borgida, Calvanese & Rodriguez-Muro 2008, ODBASE], [Calvanese, Ortiz, Simkus & Stefanoni 2013, JAIR], [Croce & Lenzerini 2018, EKAW], [Bourgaux & Ozaki 2019, AAAI], [Calvanese, Lanti, Ozaki, Peñaloza & Xiao 2019, IJCAI], [Ceylan, Lukasiewicz & al 2020, ECAI; 2021, AAAI]

SAPIENZA UNIVERSITÀ DI ROMA

Extensions to basic query answering

- Bag semantics, aggregation operators, and counting [Nikolaou, Kostylev, Konstantinidis, Kaminski, Cuenca Grau & Horrocks 2019, AIJ], [Bienvenu, Manière & Thomazo 2020, IJCAI], [Calvanese, Corman, Lanti & Razniewski 2020, IJCAI]
- Explanation and provenance [Borgida, Calvanese & Rodriguez-Muro 2008, ODBASE], [Calvanese, Ortiz, Simkus & Stefanoni 2013, JAIR], [Croce & Lenzerini 2018, EKAW], [Bourgaux & Ozaki 2019, AAAI], [Calvanese, Lanti, Ozaki, Peñaloza & Xiao 2019, IJCAI], [Ceylan, Lukasiewicz & al 2020, ECAI; 2021, AAAI]
- Inconsistency tolerant query answering and data cleaning [Lembo & Ruzzi 2007, RR], [Rosati & al 2011, DL] [Lembo, Lenzerini, Rosati, Ruzzi & Savo 2015, JWebSem], [Bienvenu & Bourgaux 2016, RW], [Bienvenu, Bourgaux & Goasdoué 2019, JAIR], [Lukasiewicz & al 2022, AIJ]

Extensions to basic query answering

- Bag semantics, aggregation operators, and counting [Nikolaou, Kostylev, Konstantinidis, Kaminski, Cuenca Grau & Horrocks 2019, AIJ], [Bienvenu, Manière & Thomazo 2020, IJCAI], [Calvanese, Corman, Lanti & Razniewski 2020, IJCAI]
- Explanation and provenance [Borgida, Calvanese & Rodriguez-Muro 2008, ODBASE], [Calvanese, Ortiz, Simkus & Stefanoni 2013, JAIR], [Croce & Lenzerini 2018, EKAW], [Bourgaux & Ozaki 2019, AAAI], [Calvanese, Lanti, Ozaki, Peñaloza & Xiao 2019, IJCAI], [Ceylan, Lukasiewicz & al 2020, ECAI; 2021, AAAI]
- Inconsistency tolerant query answering and data cleaning [Lembo & Ruzzi 2007, RR], [Rosati & al 2011, DL] [Lembo, Lenzerini, Rosati, Ruzzi & Savo 2015, JWebSem], [Bienvenu & Bourgaux 2016, RW], [Bienvenu, Bourgaux & Goasdoué 2019, JAIR], [Lukasiewicz & al 2022, AIJ]
- . . . and many others, e.g., finite model reasoning [Rosati 2008, ESWC], view-based query answering [Calvanese & al JCSS 2012], query inseparability [Konev, Kontchakov, Ludwig, Schneider, Wolter & Zakharyaschev 2011, AAAI], epistemic queries [Calvanese & al 2007, IJCAI]. . .

Outline of the talk

Many other types of reasoning services ...

Query answering is not the only reasoning task characerizing OBDM. Here we discuss:

- Metamodeling and metaquerying
- Data quality assessment
- Data abstraction
- Logical characterization of data sets

SAPIENZA
UNIVERSITÀ DI ROMA

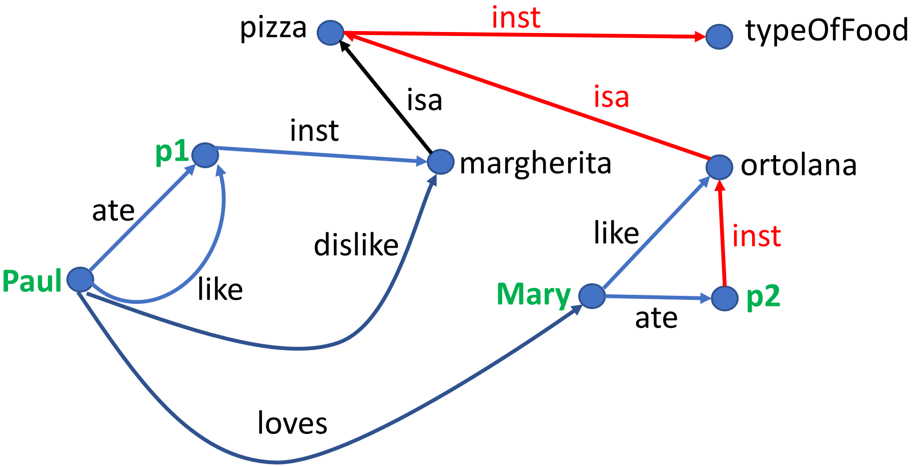
Metamodeling and metaquerying

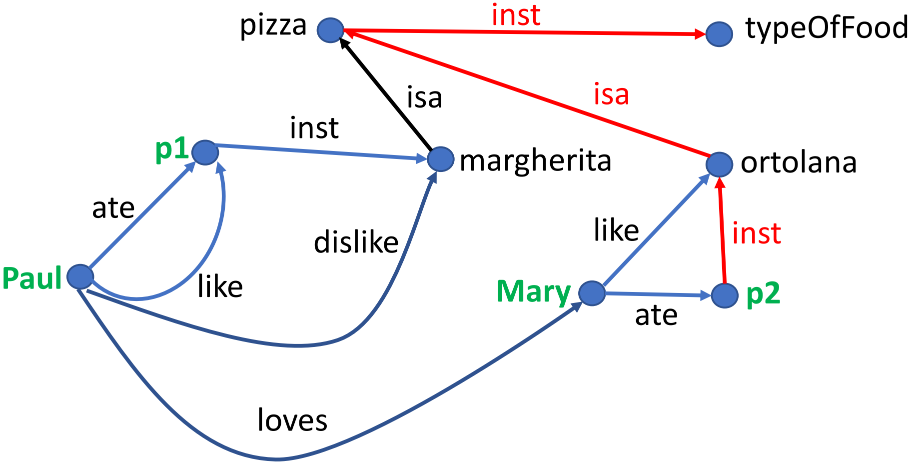
Up to now, we have assumed that the TBox and the ABox were first-order, with a strict separation between individuals and classes/relations.

- Metamodeling: specifying
  - metaclasses (classes whose instances can be themselves classes), and
  - metaproperties (relationships between metaclasses)

- Metaquerying: expressing queries with
  - variables both in predicate and object position, and
  - TBox atoms

DL-Lite and knowledge graphs





Meta-query:

$\{\ (x, z, v) \mid \texttt{ate}(x, y), z(y), z \sqsubseteq \texttt{pizza}, \texttt{ate}(x_1, y_1), z_1(y_1), z_1 \sqsubseteq \texttt{pizza}, w(x, x_1), w \sqsubseteq \neg\texttt{ate}\ \}$

The "metagrounding" technique

Let Q be a query over an ontology $\mathcal{O}$.

- a metagrounding of Q is a query Q' obtained from Q by substituting the metavariables occurring in Q in class or relation with a class or relation expression over $\mathcal{O}$, respectively
  - e.g., if $\mathcal{O}_1$ contains the classes A, B, C and the relation R, and if Q is the query

    $$Q_1() \leftarrow A \sqsubseteq \neg x, B(y), R(x,z), z(y)$$

    then one possible metagrounding of Q is the query Q' obtained by applying the substitution $\{x \leftarrow C, z \leftarrow D\}$, i.e.,

    $$Q_1() \leftarrow A \sqsubseteq \neg C, B(y), R(C,D), D(y)$$

- Answering Q through metagrounding means computing the certain answers to the union of all the metagroundings of Q

Does metagrounding work?

Example

- $\mathcal{O}_1 : \{A \sqsubseteq \neg C, R(C, A), R(B, C), C(F), B(F)\}$
- $Q_1() \leftarrow R(x, z), z(y), B(y), A \sqsubseteq \neg x$

Although no metagrounding of Q_1 gets the certain answer "true", one can show that the certain answer to Q_1 is indeed true, by partitioning the models of $\mathcal{O}_1$ into

1. those for which A and B are disjoint, and
2. those for which A and B are not disjoint (and therefore they share at least one element E)

and showing that the metagrounding $(x \leftarrow B, z \leftarrow C, y \leftarrow F)$ makes Q_1 true in (1), and the metagrounding $(x \leftarrow C, z \leftarrow A, y \leftarrow E)$ makes Q_1 true in (2).

Metagrounding does not suffice

In general, answering metaqueries cannot be done through metagrounding. Note that in the above example, the "culprit" is the uncertainty of the axiom $A \sqsubseteq \neg B$.

SAPIENZA Università di Roma

A new source of complexity

Ontology complexity: complexity wrt the part of the graph with TBox axioms.

Complexity for CQs with meta-query features in *DL-Lite* extended with metamodeling

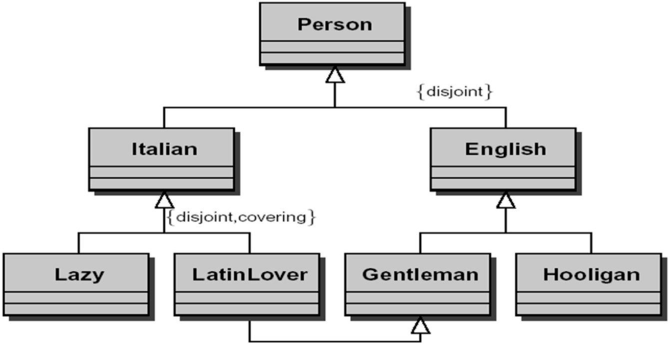
	Data complexity	**Ontology complexity**	**Combined complexity**
TBox-complete ontologies	LogSpace	PTime	NP-complete
General ontologies	LogSpace	coNP-complete	Π_2^p-complete

Quality assessment

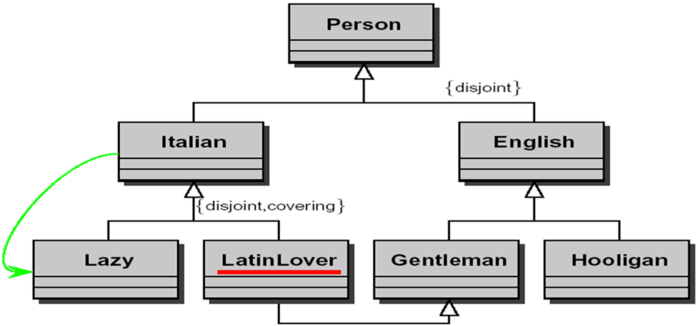
At least two types of quality checking, both based on the ontology:

- Checking the quality of intensional representation (conceptual schema)
- Checking the quality of the extensional level (data items)

Checking the quality of the conceptual schema



Checking the quality of the conceptual schema



implies

LatinLover $= \emptyset$

Italian $\subseteq$ Lazy

Italian $\equiv$ Lazy

Abstraction

Introduced in [Cima 2017, Lutz et al. 2018, Cima et al. 2019].

Definition

Given an OBDM specification $\mathcal{J} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$, a query $q_{\mathcal{S}}$ over $\mathcal{S}$, and a query $q_{\mathcal{O}}$ over $\mathcal{O}$, $q_{\mathcal{O}}$ is a **perfect** $\mathcal{J}$-abstraction of $q_{\mathcal{S}}$, if for every $\mathcal{S}$-database D such that $(\mathcal{J}, D)$ is consistent, we have that

$$q_{\mathcal{S}}^{D} = \text{cert}(q_{\mathcal{O}}, \mathcal{J}, D)$$

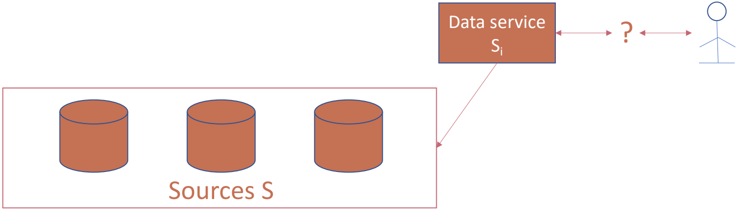
Basic computational problems:

- Verification: check if a given $q_{\mathcal{O}}$ is a perfect $\mathcal{J}$-abstraction of $q_{\mathcal{S}}$
- Existence: check whether a perfect $\mathcal{J}$-abstraction of $q_{\mathcal{S}}$ exists
- Computation: compute the perfect $\mathcal{J}$-abstraction of $q_{\mathcal{S}}$

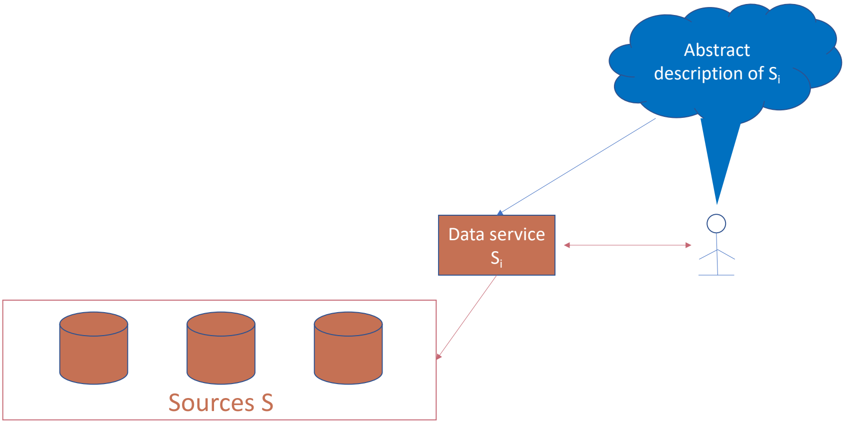
$\forall D \; q_{\mathcal{S}}^{D} = \text{cert}(q_{\mathcal{O}}, \mathcal{J}, D)$	
Query answering	**Query abstraction**
input: $q_{\mathcal{O}}$ **output**: $q_{\mathcal{S}}$	**input**: $q_{\mathcal{S}}$ **output**: $q_{\mathcal{O}}$

SAPIENZA UNIVERSITÀ DI ROMA

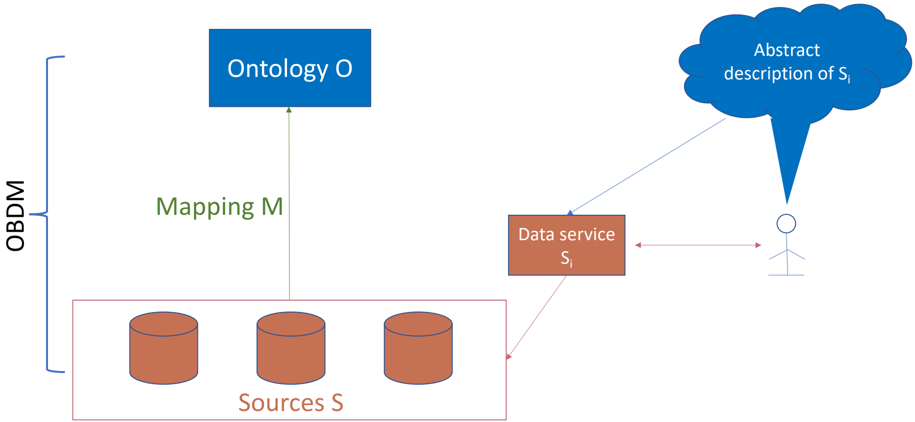
Abstraction: exploiting ontologies for explaining data service semantics



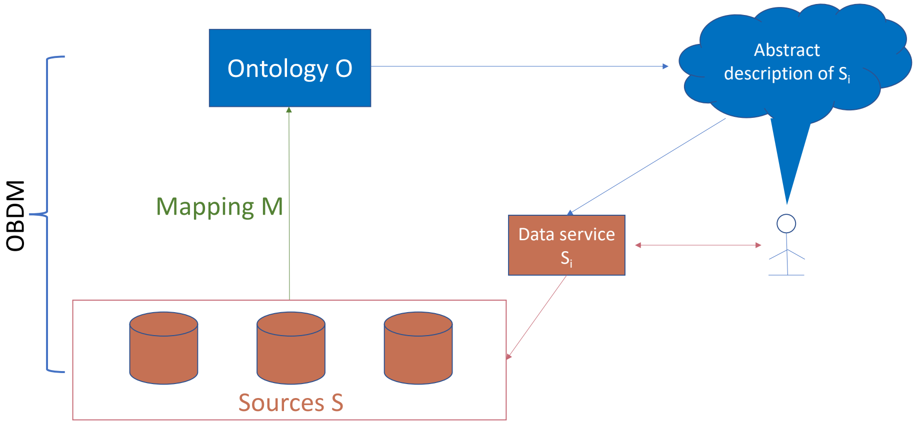
Abstraction: exploiting ontologies for explaining data service semantics



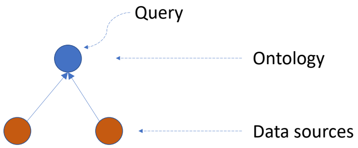
Abstraction: exploiting ontologies for explaining data service semantics



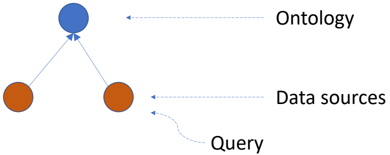
Abstraction: exploiting ontologies for explaining data service semantics



Answering vs. abstraction



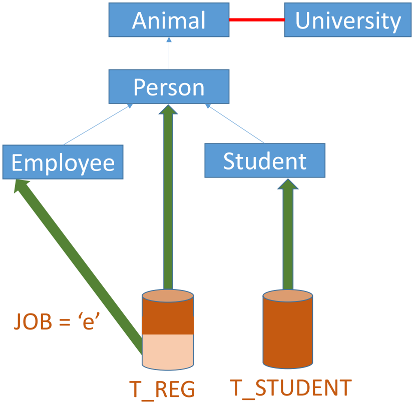
1. Query answering (from $q_{\mathcal{O}}$ to $q_{\mathcal{S}}$):
   - Extract, analyze, explore source information by accessing the ontology
2. Query abstraction (from $q_{\mathcal{S}}$ to $q_{\mathcal{O}}$):
   - Explain the content (semantics) of a data source in terms of the ontology
   - Find the best way in which (or, verify if) a given data service expressed over the data sources can be expressed in terms of the ontology
   - Automatically associate semantics to open data sets characterized by a source query



Perfect abstraction - example

$\text{Employee}(x) \rightarrow \text{Person}(x)$
$\text{Student}(x) \rightarrow \text{Person}(x)$
$\text{Person}(x) \rightarrow \text{Animal}(x)$
$\text{Animal}(x) \rightarrow \neg\ \text{University}(x)$

T_REG(x,'e') → Employee(x)
T_REG(x,y) → Person(x)
T_STUDENT(x,y,z) → Student(x)



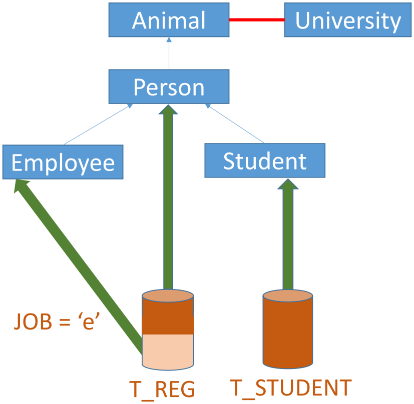
Source query q_S :
select ID as x from T_STUDENT

Source query q'_S :
select ID as x from T_REG

Perfect abstraction - example

Employee(x) $\rightarrow$ Person(x)
Student(x) $\rightarrow$ Person(x)
Person(x) $\rightarrow$ Animal(x)
Animal(x) $\rightarrow \neg$ University(x)

T_REG(x,'e') $\rightarrow$ Employee(x)
T_REG(x,y) $\rightarrow$ Person(x)
T_STUDENT(x,y,z) $\rightarrow$ Student(x)



Source query q_S :
select ID as x from T_STUDENT

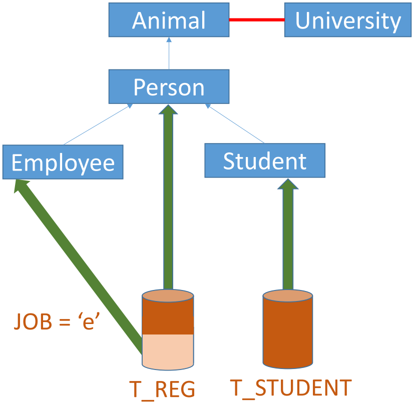
Source query q'_S :
select ID as x from T_REG

- perfect $\mathcal{J}$-abstraction of q_S: **???**

Perfect abstraction - example

Employee(x) $\rightarrow$ Person(x)
Student(x) $\rightarrow$ Person(x)
Person(x) $\rightarrow$ Animal(x)
Animal(x) $\rightarrow \neg$ University(x)

T_REG(x,'e') $\rightarrow$ Employee(x)
T_REG(x,y) $\rightarrow$ Person(x)
T_STUDENT(x,y,z) $\rightarrow$ Student(x)



Source query q_S :
select ID as x from T_STUDENT

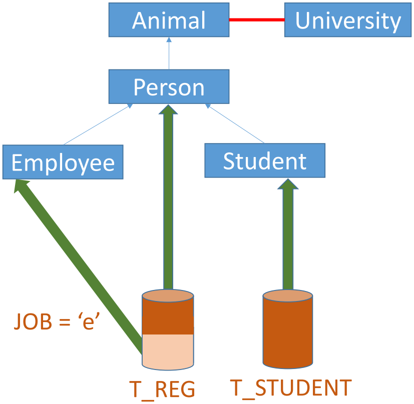
Source query q'_S :
select ID as x from T_REG

- perfect $\mathcal{J}$-abstraction of q_S: Student(x)

Perfect abstraction - example

$\text{Employee}(x) \rightarrow \text{Person}(x)$
$\text{Student}(x) \rightarrow \text{Person}(x)$
$\text{Person}(x) \rightarrow \text{Animal}(x)$
$\text{Animal}(x) \rightarrow \neg\ \text{University}(x)$

T_REG(x,'e') $\rightarrow$ Employee(x)
T_REG(x,y) $\rightarrow$ Person(x)
T_STUDENT(x,y,z) $\rightarrow$ Student(x)



Source query q_S :
select ID as x from T_STUDENT

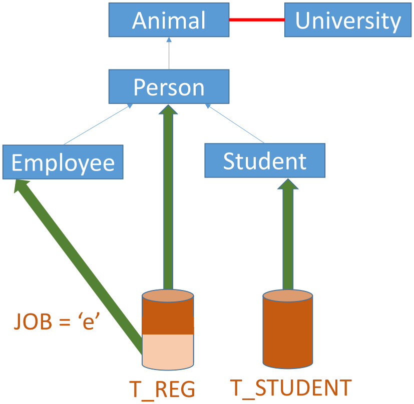
Source query q'_S :
select ID as x from T_REG

- perfect $\mathcal{J}$-abstraction of q_S: $\text{Student}(x)$
- perfect $\mathcal{J}$-abstraction of q'_S: **???**

Perfect abstraction - example

$\text{Employee}(x) \rightarrow \text{Person}(x)$
$\text{Student}(x) \rightarrow \text{Person}(x)$
$\text{Person}(x) \rightarrow \text{Animal}(x)$
$\text{Animal}(x) \rightarrow \neg\ \text{University}(x)$

T_REG(x,'e') $\rightarrow$ Employee(x)
T_REG(x,y) $\rightarrow$ Person(x)
T_STUDENT(x,y,z) $\rightarrow$ Student(x)



Source query q_S :
select ID as x from T_STUDENT

Source query q'_S :
select ID as x from T_REG

- perfect $\mathcal{J}$-abstraction of q_S: $\text{Student}(x)$
- perfect $\mathcal{J}$-abstraction of q'_S: none

On the existence of perfect abstraction

The perfect abstraction for a source query may not exist and it is in general undecidable to check if it exists.

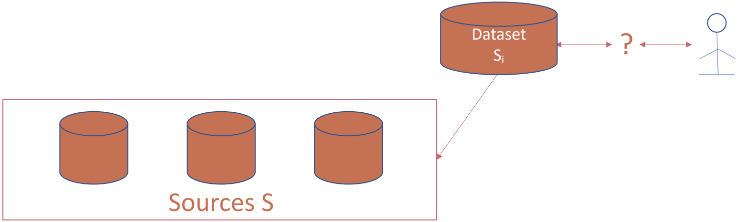
- Relaxing the definition of perfect abstraction (sound or complete abstractions)
- Looking for (sound or complete) abstractions that are the “best” in a certain class of queries
  - UCQ
  - Monotone queries

Logical characterizations of datasets in OBDM

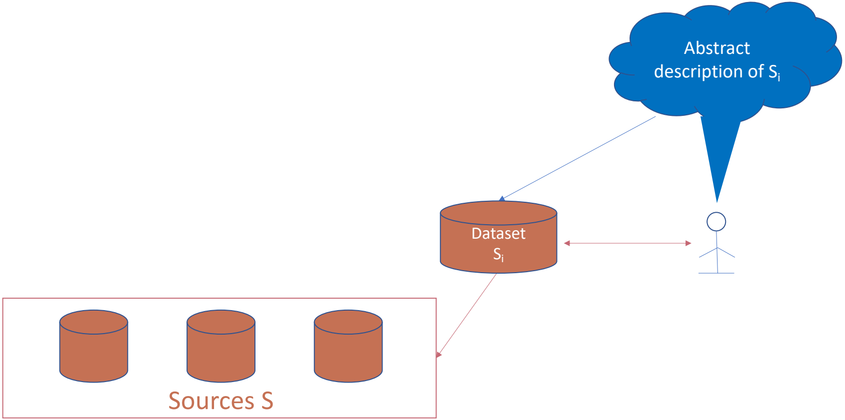
Given an OBDM system $(\mathcal{J}, D)$, and a dataset (set of tuples) λ in D, can we find a query over $\mathcal{O}$ that precisely describes ("characterizes") λ in terms of $\mathcal{O}$?

$\forall D\ q_{\mathcal{S}}^{D} = \text{cert}(q_{\mathcal{O}}, \mathcal{J}, D)$		$\text{cert}(q_{\mathcal{O}}, \mathcal{J}, D) = \lambda$
Query answering	Query abstraction	**Dataset characterization**
input: $q_{\mathcal{O}}$ **output**: $q_{\mathcal{S}}$	**input**: $q_{\mathcal{S}}$ **output**: $q_{\mathcal{O}}$	**input**: D, set of tuples λ **output**: $q_{\mathcal{O}}$

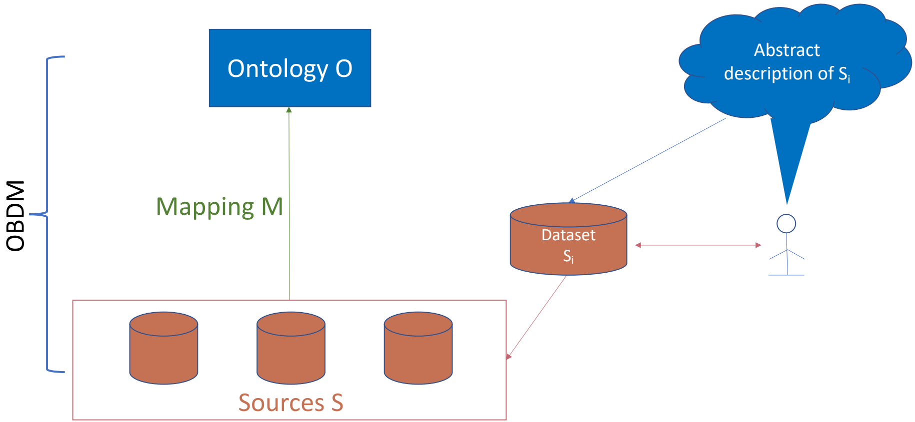
Characterization: exploiting ontologies for describing dataset semantics



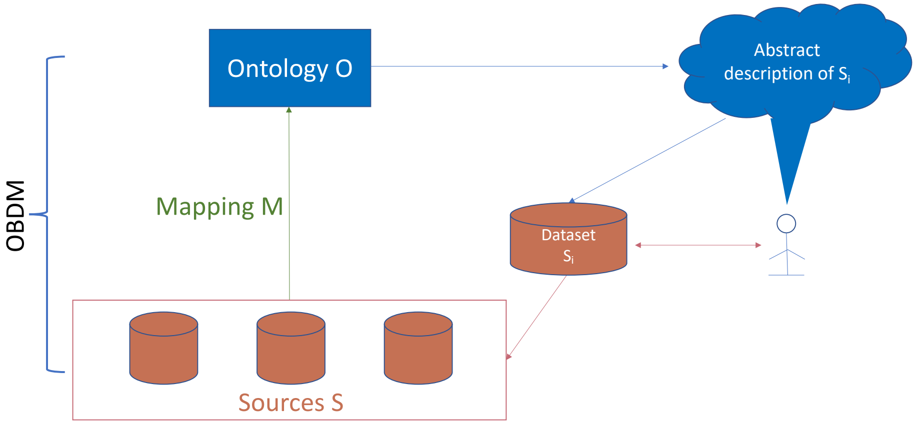
Characterization: exploiting ontologies for explainingdescribing dataset semantics



Characterization: exploiting ontologies for describing dataset semantics



Characterization: exploiting ontologies for describing dataset semantics



Logical characterization of datasets in OBDM

Studied in [Li & al 2015, PVLDB], [Basulto & al 2018, IJCAI], [Ortiz 2019, GCAI], [Cima & al 2021, CIKM] and many others

Definition (Formal definition)

Given an OBDM system $(\mathcal{J}, D)$, and a dataset (set of tuples) λ, the query $q_{\mathcal{O}} \in \mathcal{Q}$ is a **perfect $\mathcal{J}$-characterization** in the query language $\mathcal{Q}$ of λ, if

$$\text{cert}(q_{\mathcal{O}}, \mathcal{J}, D) = \lambda$$

Applications:

- Concept learning in description logic (DL): automatically construct a concept description from instances
- Reverse engineering of database queries: find a query from example answers
- Generating referring expression: find a formula that separates a single positive data item from all other data items and can thus be used as a uniquely identifying description of the data item
- Explanation for a black-box classifier: if λ is the set of tuples classified positively by a black-box model (or used as training set), then the perfect characterization of λ provides a global post-hoc explanations in terms of $\mathcal{O}$ of such a model (or, training set).

Conclusions

The present

- Logical reasoning crucial in data modeling and knowledge representation
- AI is not only machine learning: it can support domain modeling, data management, query answering, semantic data integration, data preparation, ...
- Several scenarios where ontologies/OBDM/knowledge graphs are popular (Bioinformatics, Healthcare, Open Government, Finance, Enterprise modeling, Domain modeling, NLP, data interoperability, open data publishing, ...)
- Research on algorithms and tools for reasoning about structured KR still active (see ONTOP (ONTOPIC), MASTRO (OBDA Systems), STARDOG, ...)

Many open research questions, such as

- More powerful query mechanisms (e.g., non-monotonic, negation, aggregation, privacy preserving QA, ...)
- Combine deductive reasoning and machine learning (knowledge graph embedding very popular)
- The role of KR in explainable machine learning
- The role in Data-centric AI (the discipline of systematically engineering the data, including expressing domain knowledge, needed to successfully build a machine learning system)

Credits

Joint work with(*)

- Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Antonella Poggi, Riccardo Rosati, *and many others* (OBDM and query answering)
- Antonella Poggi (meta-modeling)
- Marco Console (quality assessment)
- Gianluca Cima (data abstraction)
- Federico Croce (characterization of datasets)

Thank you for your attention!

(*) *My heart is not weary, it's light and it's free; I've got nothin' but affection for all those who've sailed with me – Bob Dylan*

SAPIENZA UNIVERSITÀ DI ROMA

References I

[1] W. A. Woods & R. J. Brachman, eds. *Research in Natural Language Understanding*. Quarterly Progress Report No. 1, BBN Report No. 3742. Cambridge, Mass.: Bolt, Beranek and Newman Inc., 1977.

[2] R. J. Brachman & H. J. Levesque. "The Tractability of Subsumption in Frame-Based Description Languages". In: *Proc. of the 4th Nat. Conf. on Artificial Intelligence (AAAI)*. 1984, pp. 34–37.

[3] A. Borgida, R. J. Brachman, D. L. McGuinness & L. A. Resnick. "CLASSIC: A Structural Data Model for Objects". In: *Proc. of the 10th ACM Int. Conf. on Management of Data (SIGMOD)*. 1989, pp. 59–67.

[4] P. F. Patel-Schneider, D. L. McGuinness, R. J. Brachman, L. A. Resnick & A. Borgida. "The CLASSIC Knowledge Representation System: Guiding Principles and Implementation Rational". In: *SIGART Bull.* 2.3 (1991), pp. 108–113.

[5] M. Schmidt-Schauss & G. Smolka. "Attributive Concept Descriptions with Complements". In: *Artificial Intelligence* 48.1 (1991), pp. 1–26.

References II

[6] F. Baader. "Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles". In: *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. 1991.

[7] B. Nebel. "Terminological Cycles: Semantics and Computational Properties". In: *Principles of Semantic Networks*. Ed. by J. F. Sowa. Morgan Kaufmann, 1991, pp. 331–361.

[8] F. M. Donini, M. Lenzerini, D. Nardi & W. Nutt. "The Complexity of Concept Languages". In: *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*. 1991, pp. 151–162.

[9] K. Schild. "A Correspondence Theory for Terminological Logics: Preliminary Report". In: *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. 1991, pp. 466–471.

[10] G. De Giacomo. "Decidability of Class-Based Knowledge Representation Formalisms". PhD thesis. Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1995.

[11] I. Horrocks. "Using an Expressive Description Logic: FaCT or Fiction?" In: *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*. 1998, pp. 636–647.

References III

[12] I. Horrocks, U. Sattler & S. Tobies. “Practical Reasoning for Very Expressive Description Logics”. In: *J. of the Interest Group in Pure and Applied Logic* 8.3 (2000), pp. 239–264.

[13] D. Berardi, D. Calvanese & G. De Giacomo. “Reasoning on UML Class Diagrams”. In: *Artificial Intelligence* 168.1–2 (2005), pp. 70–118.

[14] D. Calvanese, G. De Giacomo & M. Lenzerini. “On the Decidability of Query Containment under Constraints”. In: *Proc. of the 17th ACM Symp. on Principles of Database Systems (PODS)*. 1998, pp. 149–158.

[15] C. Nikolaou, E. V. Kostylev, G. Konstantinidis, M. Kaminski, B. Cuenca Grau & I. Horrocks. “Foundations of Ontology-Based Data Access Under Bag Semantics”. In: *Artificial Intelligence* 274 (2019), pp. 91–132.

[16] M. Bienvenu, Q. Manière & M. Thomazo. “Answering Counting Queries over DL-Lite Ontologies”. In: *Proc. of the 29th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. Ed. by C. Bessiere. IJCAI Org., 2020, pp. 1608–1614.

[17] D. Calvanese, J. Corman, D. Lanti & S. Razniewski. “Counting Query Answers over a DL-Lite Knowledge Base”. In: *Proc. of the 29th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. Ed. by C. Bessiere. IJCAI Org., 2020, pp. 1658–1666.

References IV

[18] A. Borgida, D. Calvanese & M. Rodriguez-Muro. "Explanation in the *DL-Lite* Family of Description Logics". In: *Proc. of the 7th Int. Conf. on Ontologies, DataBases, and Applications of Semantics (ODBASE)*. Vol. 5332. Lecture Notes in Computer Science. Springer, 2008, pp. 1440–1457.

[19] D. Calvanese, M. Ortiz, M. Simkus & G. Stefanoni. "Reasoning about Explanations for Negative Query Answers in *DL-Lite*". In: *J. of Artificial Intelligence Research* 48 (2013), pp. 635–669.

[20] C. Bourgaux & A. Ozaki. "Querying Attributed DL-Lite Ontologies Using Provenance Semirings". In: *Proc. of the 33rd AAAI Conf. on Artificial Intelligence (AAAI)*. AAAI Press, 2019, pp. 2719–2726.

[21] D. Calvanese, D. Lanti, A. Ozaki, R. Peñaloza & G. Xiao. "Enriching Ontology-based Data Access with Provenance". In: *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. IJCAI Org., 2019, pp. 1616–1623.

[22] D. Lembo & M. Ruzzi. "Consistent Query Answering over Description Logic Ontologies". In: *Proc. of the 1st Int. Conf. on Web Reasoning and Rule Systems (RR)*. 2007.

[23] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi & D. F. Savo. "Inconsistency-tolerant Query Answering in Ontology-based Data Access". In: *J. of Web Semantics* 33 (2015), pp. 3–29.

References V

[24] M. Bienvenu & C. Bourgaux. “Inconsistency-Tolerant Querying of Description Logic Knowledge Bases”. In: *Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering – 12th Int. Summer School Tutorial Lectures (RW)*. Ed. by J. Z. Pan, D. Calvanese, T. Eiter, I. Horrocks, M. Kifer, F. Lin & Y. Zhao. Vol. 9885. Lecture Notes in Computer Science. Springer, 2016, pp. 156–202.

[25] M. Bienvenu, C. Bourgaux & F. Goasdoué. “Computing and Explaining Query Answers over Inconsistent DL-Lite Knowledge Bases”. In: *J. of Artificial Intelligence Research* 64 (2019), pp. 563–644.

[26] R. Rosati. “Finite Model Reasoning in *DL-Lite*”. In: *Proc. of the 5th European Semantic Web Conf. (ESWC)*. Vol. 5021. Lecture Notes in Computer Science. Springer, 2008, pp. 215–229.

[27] B. Konev, R. Kontchakov, M. Ludwig, T. Schneider, F. Wolter & M. Zakharyaschev. “Conjunctive Query Inseparability of OWL 2 QL TBoxes”. In: *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI)*. 2011, pp. 221–226.