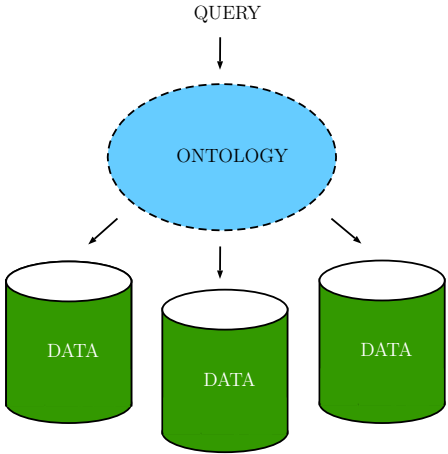
Ontology-Mediated Query Answering

Federico Ulliana

Université de Montpellier, LIRMM, INRIA GraphIK

Joint work with
Marie-Laure Mugnier, Marie-Christine Rousset, Michel Leclère,
Meghyn Bienvenu, Pierre Bourhis, Sophie Tison

Ontology-Mediated Query Answering



Ontological knowledge :

1. Enriched query vocabulary
2. Incomplete information
3. Unified view of data

Applications : Data-Integration, Semantic Web

Ontology-Mediated Query Answering

Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?"

Ontology-Mediated Query Answering

Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?" ✗

Ontology-Mediated Query Answering

Know. : "Every **Professor** has a **Contact**" Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?"

Ontology-Mediated Query Answering

Know. : "Every **Professor** has a **Contact**" Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?" ✓

Ontology-Mediated Query Answering

Know. : "Every **Professor** has a **Contact**" Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?" ✓

This approach applies to scientific data (e.g., medicine, biology, chemistry), bibliographical data, governmental (open) data

Knowledge Base := (Database,Ontology)

Ontologies model (some aspects of) the world

- Introduce vocabulary relevant to a domain
  - (typically) Classes : **Professor**, **Contact** Relations : is a, has a
    Hierarchies : **Professor** $\leq$ **Faculty** $\leq$ **Person**

- Formally specifies (rich) semantics and meaning of the vocabulary
  - (typically) in terms of Rules
    "Every **Professor** has a **Contact**"
    "**Professors** are disjoint from **Students**"
    "Every **Class** it taught by a single **Professor**"

Ontologies vs. Database Schemas

Ontology $\sim$ DB schema : describe structure and constraints on data

Schema define legal DB states

Ontologies entail implicit facts

DB schemas do not influence query answering (beside optim.), ontologies do.

Closed world assumption (CWA)

- Missing information $\sim$ false

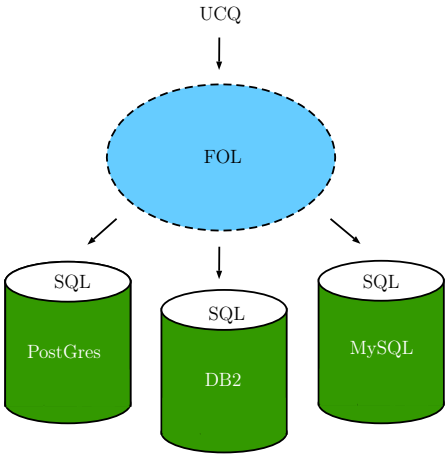
Open world assumption (OWA)

- Missing inform. $\sim$ unknown

OMQA

- Relational Data
- RDF Data (not covered today)
- NOSQL Data

OMQA for Relational Databases



- Relational technology ubiquitous

  Boost for OMQA investigations

- Ontologies : Description Logics, Existential Rules

- Why this marriage works so well ?

Logical View of Databases

contact

alice	a@lirmm.fr
alice	0405060708
bob	z_0

permanent

alice
charles

$$\exists z_0 \big(\; contact(alice, a@lirmm.fr) \wedge permanent(alice) \wedge contact(alice, 0405060708) \wedge permanent(charles) \wedge contact(bob, z_0) \; \big)$$

$\pi_{[1]}(\texttt{contact}) \bowtie \texttt{permanent}$

$\pi_{\emptyset}(\sigma_{[1=\texttt{bob}]}\texttt{contact})$

$$Q(x) :- \exists x, y \, \big(contact(x, y) \wedge permanent(x) \big)$$

$$Q_{bool}() :- \exists x \, \big(contact(bob, x) \big)$$

Theorem (Codd'72)

Relational Algebra $\approx$ *FO*

Homomorphism, Entailment, and Query Answering

Variable substitution $h : Q \longrightarrow F$

$Q(x) :- \exists x, y \left(contact(x, y) \wedge permanent(x)\right)$ $\quad Q_{bool}() :- \exists x \left(contact(bob, x)\right)$

$h_1 = \{x \mapsto alice, y \mapsto a@lirmm.fr\}$ $\quad h_3 = \{x \mapsto z_0\}$

$h_2 = \{x \mapsto alice, y \mapsto 0405060708\}$

$Answ(Q, F) = \{(alice)\}$ $\quad Answ(Q_{bool}, F) =$ "Yes"

$$F \models Q$$

Ontology Languages

Description Logics

- DL-Lite [?]
- EL [?]
- Horn DLs [?]
- Foundations of OWL 2 Profiles [?]

Positive Existential Rules [?]

- Same as Tuple Generating Dependencies (TGDs) [?]
- See also Datalog+/- [?]
- See Conceptual Graphs [?, ?]
- Generalize Horn DLs

Positive Existential Rules

$$\forall \vec{x}, \vec{y} \left(\mathsf{Body}(\vec{x}, \vec{y}) \longrightarrow \exists \vec{z}\ \mathsf{Head}(\vec{y}, \vec{z}) \right)$$

Body, Head positive conjunctions of atoms, without function symbols.

$$\forall x \Big(\mathsf{Professor}(x) \longrightarrow \exists Z\ \mathsf{hasContact}(x, Z) \wedge \mathsf{Contact}(Z) \Big)$$

Existentially quantified variables enable "value invention".

Generalize DL-Lite / OWL 2 QL and RDF(S)

DL-Lite Axioms	Existential Rules
$A \sqsubseteq B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \mathsf{B}(x)\big)$
$A \sqsubseteq \exists R$	$\forall x\big(\mathsf{A}(x) \longrightarrow \exists z.\mathsf{R}(x,z)\big)$
$\exists R \sqsubseteq A$	$\forall xy\big(\mathsf{R}(x,y) \longrightarrow \mathsf{A}(x)\big)$
$\exists R \sqsubseteq \exists P$	$\forall xy\big(\mathsf{R}(x,y) \longrightarrow \exists z.\mathsf{P}(x,z)\big)$
$A \sqsubseteq \exists R.B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \exists z.\mathsf{R}(x,z) \wedge \mathsf{B}(z)\big)$
$P \sqsubseteq Q$	$\forall xy\big(\mathsf{P}(x,y) \longrightarrow \mathsf{Q}(x,y)\big)$
$A \sqsubseteq \neg B$	$\forall x\big(\mathsf{A}(x) \wedge \mathsf{B}(x) \longrightarrow \bot\big)$

Generalize EL Description Logics / OWL 2 EL

EL-Lite Axioms	Existential Rules
$A \sqsubseteq B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \mathsf{B}(x)\big)$
$A \sqcap B \sqsubseteq C$	$\forall x\big(\mathsf{A}(x) \wedge B(x) \longrightarrow \mathsf{C}(x)\big)$
$A \sqsubseteq \exists R.B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \exists z.\mathsf{R}(x,z) \wedge B(z)\big)$
$\exists R.B \sqsubseteq A$	$\forall xy\big(\mathsf{R}(x,y) \wedge \mathsf{B}(y) \longrightarrow \mathsf{A}(x)\big)$

Rule (Model) Semantics

$(F \models \sigma)$ F is a model of σ when any homomorphism $h(\mathsf{Body}(\sigma)) \subseteq F$ can be extended to some $h' \supseteq h$ such that $h'(\mathsf{Head}(\sigma)) \subseteq F$

$$\sigma : \forall x \big(\mathsf{Professor}(x) \longrightarrow \exists Z.\mathsf{hasContact}(x, Z)\big)$$

$$\mathsf{Professor}(\mathit{alice}) \not\models \sigma$$

$$\mathsf{Professor}(\mathit{alice}) \wedge \mathsf{hasContact}(\mathit{alice}, \mathit{0405060708}) \models \sigma$$

Rule Application

A rule σ apply on F if there is an homomorphism $\quad h(\mathsf{Body}(\sigma)) \subseteq F$

Effective application adds $\quad h^{\mathbf{safe}}(\mathsf{Head}(\sigma))$

$$\sigma : \forall x \big(\mathsf{Professor}(x) \longrightarrow \exists Z.\mathsf{hasContact}(x, Z)\big) \qquad F : \mathsf{Professor}(alice)$$

$$h^{\mathbf{safe}} = \{\ x \mapsto alice,\ Z \mapsto z_0\ \}$$

$$\exists z_0 \big(\ \mathsf{Professor}(alice) \wedge \mathsf{hasContact}(alice, z_0)\ \big)$$

The Chase Procedure

Algorithmic tool for constructing universal models (can answer any query).

Until possible$^{(*)}$ apply the rules of Σ on the factbase F.

$$F, \Sigma \models Q \quad \Longleftrightarrow \quad \mathit{Chase}(F, \Sigma) \models Q$$

Saturation (also called materialization) approach.

Why Materializing Data ?

Pro :

- Materialization is independent from queries.

Cons :

- Possible data blowup
- Needs maintenance under updates
- Requires writing access rights to sources

Query Rewriting

Algorithmic tool for computing an equivalent of the input query Q (under Σ).

Until possible$^{(*)}$, propagate the rules of Σ into the query Q.

$$\forall x(\mathsf{Professor}(x) \longrightarrow \exists Z\ \mathsf{hasContact}(x, Z))$$

piece based unification $\updownarrow$ $\updownarrow$

$$Q' :- \mathsf{Professor}(alice) \qquad Q :- \exists w.\mathsf{hasContact}(alice, w)$$

$$Rew(Q, \Sigma) = Q' \vee Q$$

Query Rewriting

$$F, \Sigma \models Q \quad \iff \quad F \models Rew(Q, \Sigma)$$

Duality

$$Chase(F, \Sigma) \models Q \quad \iff \quad F \models Rew(Q, \Sigma)$$

Why Rewriting Queries ?

Pro :

- Rewriting is independent from data
- Resilient to updates
- Deals with read-only sources

Cons :

- Query blowup (exponential in many practical cases)
- Calls for compact rewritings (many investigations)
- Use mixed approaches

Looking for Terminating Conditions

Chase can be infinite

$$\forall x, y\big(\ \mathsf{hasParent}(x, y) \rightarrow \exists z.\mathsf{hasParent}(y, z)\ \big)$$

$$F : \mathsf{hasParent}(alice, bob) \qquad \exists z_0, z_1 \ldots \big(\mathsf{hasParent}(alice, bob) \wedge \mathsf{hasParent}(bob, z_0) \wedge \mathsf{hasParent}(z_0, z_1) \ldots\big)$$

Rewriting can be infinite

$$\forall x, y, z(\mathsf{partOf}(x, y) \wedge \mathsf{partOf}(y, z) \rightarrow \mathsf{partOf}(x, z))$$

$$Q : \mathsf{partOf}(foot, body) \qquad \bigvee_{i=0}^{n} \exists x_0 \ldots x_n\big(\mathsf{partOf}(foot, x_0) \wedge \cdots \wedge \mathsf{partOf}(x_n, body)\big)$$

Finite Expansion and Unification

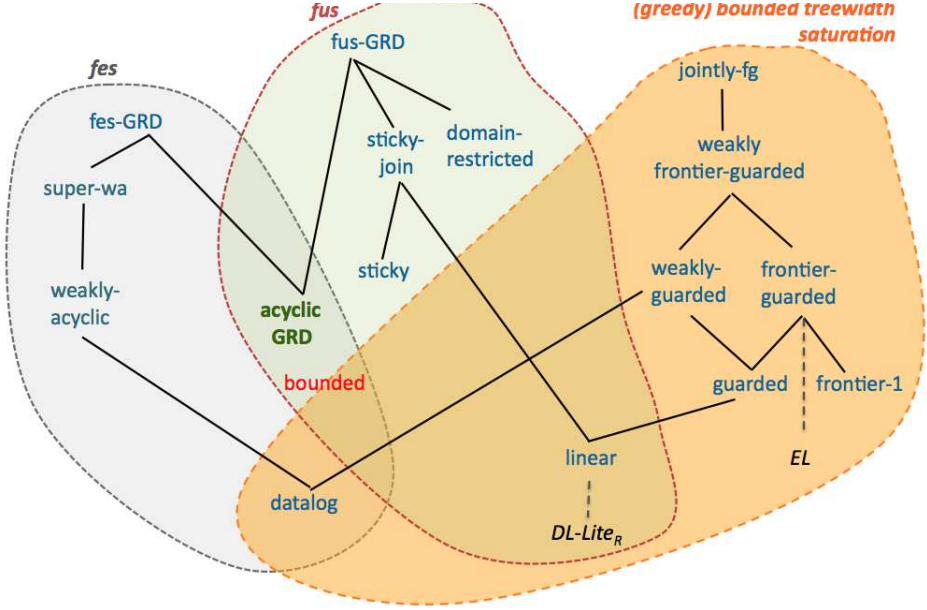
Σ finite expansion $\quad \forall F.\ \exists k.\ \mathit{Chase}^{(k)}(F, \Sigma) \equiv \mathit{Chase}^{(k+1)}(F, \Sigma)$

$$\forall x, y, z(\mathsf{partOf}(x, y) \wedge \mathsf{partOf}(y, z) \rightarrow \mathsf{partOf}(x, z))$$

Σ finite unification $\quad \forall Q.\ \exists k.\ \mathit{Rew}^{(k)}(Q, \Sigma) \equiv \mathit{Rew}^{(k+1)}(Q, \Sigma)$

$$\forall x, y\big(\mathsf{hasParent}(x, y) \rightarrow \exists z.\mathsf{hasParent}(y, z)\big)$$

Abstract classes for which query answering is decidable.


fus
(greedy) bounded treewidth saturation
fes
fus-GRD
jointly-fg
fes-GRD
sticky-join
domain-restricted
weakly frontier-guarded
super-wa
sticky
weakly-guarded
frontier-guarded
weakly-acyclic
acyclic GRD
bounded
guarded
frontier-1
linear
EL
datalog
DL-Lite$_R$

Implementations

DL-Lite

- OnTop, Rapid, Iqaros, Presto/Mastro, Requiem, HermiT

EL

- Rapid, ELK

Existential Rules / RDF Rules

- RDFOX **Graal** graphik-team.github.io/graal/