Islands of tractability in ontology-based data access

Michael Zakharyaschev

Department of Computer Science and Information Systems,
Birkbeck, University of London

http://www.dcs.bbk.ac.uk/~michael

EPSRC
Engineering and Physical Sciences
Research Council

supported by EPSRC grants ExODA EP/H05099X and iTract EP/M012670

Data access in industry

(from Norwegian Petroleum Directorate's FactPages)

show me the wellbores completed before 2008 where Statoil as a drilling operator sampled less than 10 meters of cores

5 days later:

```
SELECT DISTINCT cores.wlbName, cores.lenghtM, wellbore.wlbDrillingOperator, wellbore.wlbCompletionYear
FROM
    ( (SELECT wlbName, wlbNpdidWellbore, (wlbTotalCoreLength * 0.3048) AS lenghtM
       FROM wellbore_core
       WHERE wlbCoreIntervalUom = '(ft )'  )
      UNION
      (SELECT wlbName, wlbNpdidWellbore, wlbTotalCoreLength AS lenghtM
       FROM wellbore_core
       WHERE wlbCoreIntervalUom = '(m )'  )
    ) as cores,
    ( (SELECT wlbNpdidWellbore, wlbDrillingOperator, wlbCompletionYear
       FROM wellbore_development_all
      UNION
      (SELECT wlbNpdidWellbore, wlbDrillingOperator, wlbCompletionYear
       FROM wellbore_exploration_all  )
      UNION
      (SELECT wlbNpdidWellbore, wlbDrillingOperator, wlbCompletionYear
       FROM wellbore_shallow_all  )
    ) as wellbore
WHERE wellbore.wlbNpdidWellbore = cores.wlbNpdidWellbore
    ...
```

In STATOIL:

1,000 TB of relational data

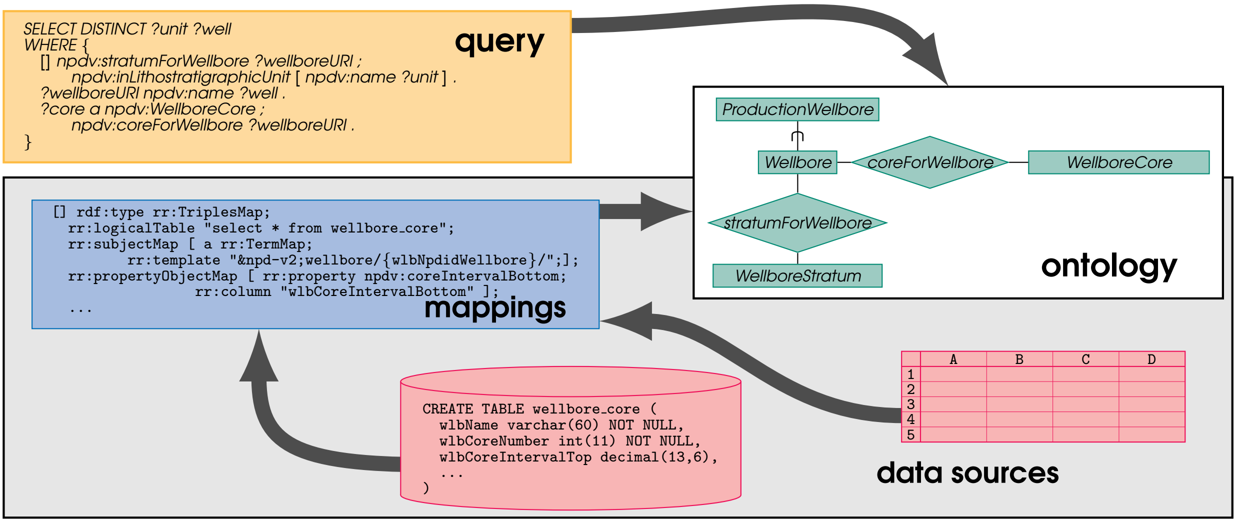
2,000 tables

different schemas

30–70% of time on data gathering

Ontology-based data access (OBDA)

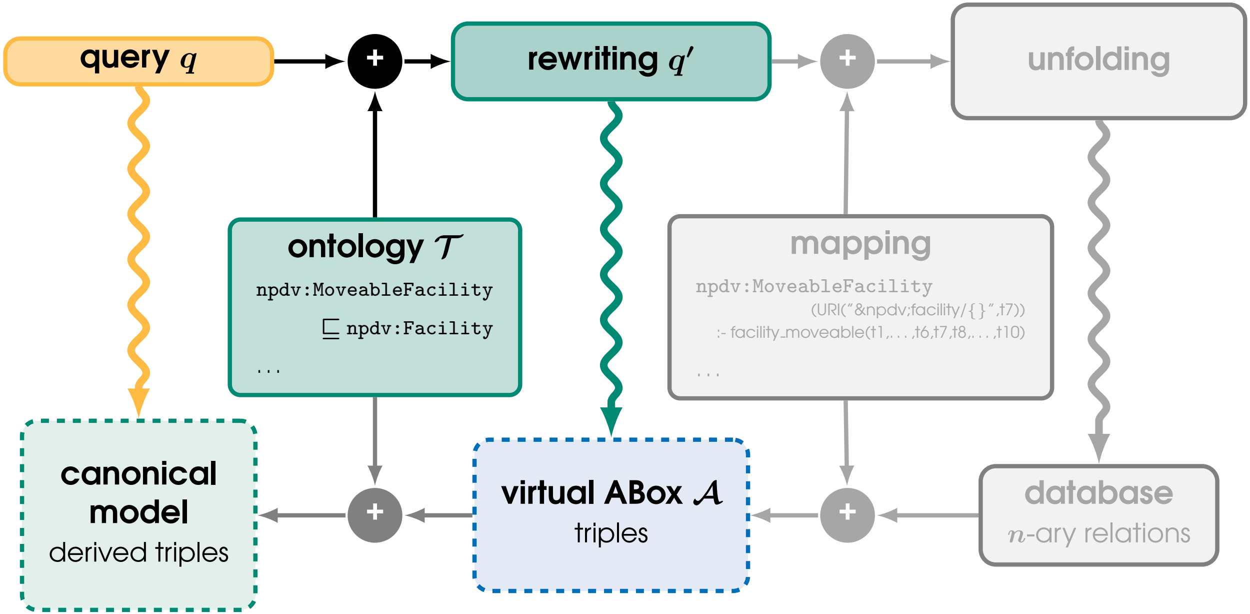
(the Romans ≈ **2007**)



Ontology

- gives a high-level conceptual view of the data
- provides a convenient & natural vocabulary for user queries
- enriches incomplete data with background knowledge

OBDA via FO-rewriting



for all $\mathcal{A}$ and $\vec{a}$, $\quad \mathcal{T}, \mathcal{A} \models q(\vec{a}) \iff \mathcal{I}_{\mathcal{A}} \models q'(\vec{a})$

reduction to DB query evaluation

OWL 2 QL profile of *OWL 2* (W3C 2012)

Roles $\varrho(x,y) \ ::= \ \top \ | \ P(x,y) \ | \ P(y,x)$ $\qquad R ::= \top \mid P \mid P^-$

Basic concepts $\tau(x) \ ::= \ \top \ | \ A(x) \ | \ \exists y\, \varrho(x,y)$ $\qquad B ::= \top \mid A \mid \exists R$

TBoxes

$\forall x\, (\tau(x) \to \tau'(x))$ $\qquad B \sqsubseteq B'$

$\forall x, y\, (\varrho(x,y) \to \varrho'(x,y))$ $\qquad R \sqsubseteq R'$

$\forall x\, \varrho(x,x)$ $\qquad R$ is reflexive

$\forall x\, (\tau(x) \land \tau'(x) \to \bot)$ $\qquad B \sqcap B' \sqsubseteq \bot$

$\forall x, y\, (\varrho(x,y) \land \varrho'(x,y) \to \bot)$ $\qquad R \sqcap R' \sqsubseteq \bot$

$\forall x\, (\varrho(x,x) \to \bot)$ $\qquad R$ is irreflexive

Sugar $\forall x\, (\tau(x) \to \exists y\, (\varrho_1(x,y) \land \dots \land \varrho_k(x,y) \land \tau'(y)))$ $\qquad B \sqsubseteq \exists R.B'$

(expressible via additional role inclusions)

ABoxes $\{A(a),\ P(a,b),\ \dots\}$

based on the '*DL-Lite* family' designed by the Romans ($\approx$ **2005**) and extended by Artale, Calvanese, Kontchakov & Z (**2007–9**)

Example

Staff ontology $\mathcal{T}$

$$\forall x\,(\textit{ProjectManager}(x) \to \exists y\,(\textit{isAssistedBy}(x,y) \land \textit{PA}(y)))$$
$$\forall x\,(\exists y\,\textit{managesProject}(x,y) \to \textit{ProjectManager}(x))$$
$$\forall x\,(\textit{ProjectManager}(x) \to \textit{Staff}(x))$$
$$\forall x\,(\textit{PA}(x) \to \textit{Secretary}(x))$$

User query q: find the staff assisted by secretaries

$$q(x) = \exists y\,(\textit{Staff}(x) \land \textit{isAssistedBy}(x,y) \land \textit{Secretary}(y)))$$

PE-rewriting of ontology-mediated query $(\mathcal{T}, q)$

$$q'(x) = \exists y\,\big[\textit{Staff}(x) \land \textit{isAssistedBy}(x,y) \land (\textit{Secretary}(y) \lor \textit{PA}(y))\big] \lor \textit{ProjectManager}(x) \lor \exists z\,\textit{managesProject}(x,z)$$

Why are *OWL 2 QL* OMQs FO-rewritable?

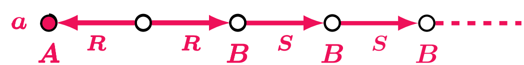
✓ **Canonical model (chase)** $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ of a given consistent $(\mathcal{T},\mathcal{A})$

homomorphically embeddable into every model of $(\mathcal{T},\mathcal{A})$

→ $\boxed{\mathcal{T},\mathcal{A} \models q \iff \mathcal{C}_{\mathcal{T},\mathcal{A}} \models q}$ for any CQ q

Example: $\mathcal{T} = \{A \sqsubseteq \exists R^-.\exists R.B, \quad B \sqsubseteq \exists S.B\}$ $\qquad \mathcal{A} = \{A(a)\}$

$\mathcal{C}_{\mathcal{T},\mathcal{A}}$



all Horn DLs have canonical models but OMQ $(\{\exists R.A \sqsubseteq A\}, A(x))$ is not FO-rewritable

(recursive datalog needed)

✓ **Bounded depth derivation property:** there is a function f such that

$\boxed{\mathcal{T},\mathcal{A} \models q \iff \mathcal{C}^N_{\mathcal{T},\mathcal{A}} \models q}$ with $\mathcal{C}^N_{\mathcal{T},\mathcal{A}}$ constructed in $N = f(|\mathcal{T}|, |q|)$ steps

$\Leftrightarrow$ **FO-rewritability**

f is **polynomial** for *OWL 2 QL*

What is the price of OBDA?

– reduction to DB query evaluation could be too expensive

→ OBDA would not be viable

1 what is the size of rewritings ?

– depending on the type of OMQs

– depending on the type of rewritings

new research (succinctness) problem

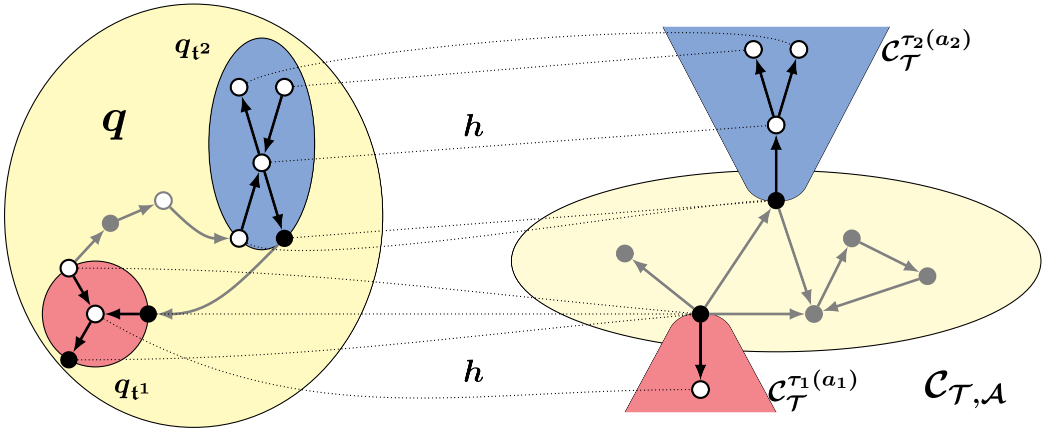
2 what is the combined complexity of OMQ answering ?

– depending on the type of OMQs

well-known problem in DB theory

it may turn out that reduction to DB query evaluation is not most optimal way of OMQ answering

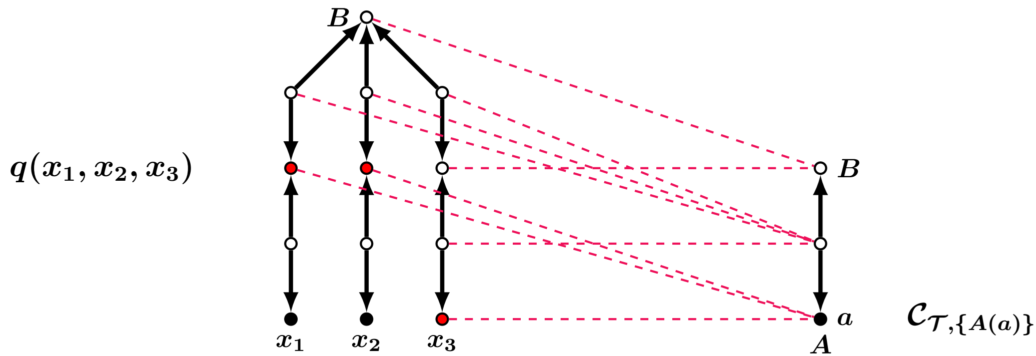
Tree-witness rewriting of OMQ $Q = (\mathcal{T}, q)$



$$q_{\mathsf{tw}}(\vec{x}) \;=\; \bigvee_{\substack{\Theta \text{ independent set} \\ \text{of tree witnesses}}} \exists \vec{y} \Big(\bigwedge_{S(\vec{z}) \in q \setminus q_\Theta} S(\vec{z}) \;\land\; \bigwedge_{\mathfrak{t} \in \Theta} \mathsf{tw}_{\mathfrak{t}} \Big)$$

Θ is **independent** if $q_{\mathfrak{t}} \cap q_{\mathfrak{t}'} = \emptyset$, for any distinct $\mathfrak{t}, \mathfrak{t}' \in \Theta$

The number of tree witnesses



exponentially-many tree witnesses → huge tw-rewriting

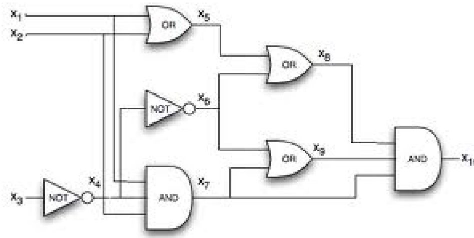
however, it can be simplified to a polynomial-size PE-rewriting:

$$q(x_1, x_2, x_3) \;\vee\; \exists z \left[A(z) \wedge \textstyle\bigwedge_{i=1}^{n} \big((x_i = z) \vee \exists y \,(R(y, x_i) \wedge R(y, z))\big)\right]$$

can we always do this?

Circuit complexity

P/poly: the class of problems decidable by **polynomial-size** circuit families



P $\subseteq$ P/poly

if NP $\not\subseteq$ P/poly then P $\neq$ NP

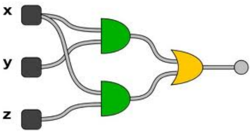
– almost all Boolean functions with n inputs require circuits of size $\Theta(2^n/n)$

(Shannon 1949)

are there **complex** Boolean functions f_n in NP**?** (known lower bound: $5n - o(n)$)

nobody knows, but ...

Monotone circuit complexity



(Razborov, Raz, *et al.* 1985)

Boolean variables e_{ij} give graph $G = (V, E)$: $V = \{1, \ldots, n\}$, $E = \{\{i, j\} \mid e_{ij} = 1\}$

– $\textsc{Clique}_{n,k}(\vec{e}) = 1$ iff G contains a **k-clique** (e.g., for $k \leq n^{1/4}$)

monotone circuits: **exp** $(2^{\varepsilon\sqrt{k}})$

monotone formulas: **exp**

formulas with ¬: **superpoly** unless NP ⊆ P/poly

– $\textsc{Matching}_n(\vec{e}) = 1$ iff the bipartite graph $\vec{e}$ with n vertices in each part has a **perfect matching** (subset of edges containing every node once)

monotone formulas: **exp**

formulas with ¬: **poly**

Tree-witness rewriting as a Boolean function

OMQ $Q = (\mathcal{T}, q)$ ⟶ a **hypergraph** $H_Q = (V, E)$ where

vertices V = atoms of q

hyperedges E = tree witnesses q_t

monotone Boolean **hypergraph function** for Q (or hypergraph H_Q)

$$f_Q = \bigvee_{E' \subseteq E \text{ independent}} \Big(\bigwedge_{v \in V \setminus V_{E'}} p_v \wedge \bigwedge_{e \in E'} p_e \Big)$$

(some tweaks required in case of exponentially-many tree witnesses)

- Boolean formula φ for f_Q ⟶ FO-rewriting of size $O(|\varphi| \cdot |Q|)$
- monotone Boolean formula φ for f_Q ⟶ PE-rewriting ———
- monotone Boolean circuit φ for f_Q ⟶ NDL-rewriting ——— (nonrecursive datalog)

tool for obtaining upper succinctness and complexity bounds

using classical circuit complexity

Tool for lower bounds

For any OMQ $Q = (\mathcal{T}, q)$ and assignment $\alpha\colon$ *predicates*$(q) \to \{0, 1\}$,

$$\mathcal{A}_\alpha = \{A(a) \mid \alpha(A) = 1\} \cup \{P(a, a) \mid \alpha(P) = 1\}$$

ABox with **a single individual a**

Primitive evaluation function: $g_Q(\alpha) = 1 \quad \Leftrightarrow \quad \mathcal{T}, \mathcal{A}_\alpha \models q(\vec{a})$

- FO-rewriting q' of Q → Boolean formula for g_Q of size $O(|q'|)$
- PE-rewriting q' of Q → monotone Boolean formula for g_Q ———
- NDL-rewriting q' of Q → monotone Boolean circuit for g_Q ———

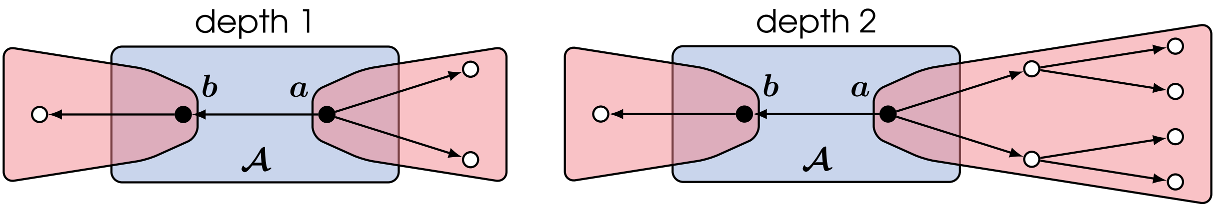
(proof by quantifier elimination)

tool for obtaining lower succinctness bounds

using classical circuit complexity

Case study: OMQs with ontologies of depth 1

no axioms such as $A \sqsubseteq \exists P, \quad \exists P^- \sqsubseteq \exists R$



$Q = (\mathcal{T}, q)$ with $\mathcal{T}$ of **depth 1** → hypergraph H_Q is of **degree** ≤ 2

each vertex belongs to ≤ 2 hyperedges

hypergraph H of degree ≤ 2 → $\exists$ OMQ Q_H with $\mathcal{T}$ of depth 1 and $H \cong H_{Q_H}$

What can hypergraph functions of degree 2 compute?

Hypergraph programs (HGPs)

An **HGP** is a hypergraph $\boldsymbol{H} = (\boldsymbol{V}, \boldsymbol{E})$ with every vertex labelled by $\mathbf{0}$, $\mathbf{1}$, $\boldsymbol{p_i}$ or $\neg \boldsymbol{p_i}$

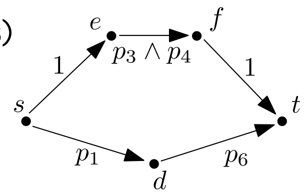
computes $\boldsymbol{f}$: $\boldsymbol{f}(\vec{\boldsymbol{\alpha}}) = \mathbf{1} \Leftrightarrow$ there is an independent $\boldsymbol{E}' \subseteq \boldsymbol{E}$ **covering all zeros**

(contains all vertices whose labels evaluate to $\mathbf{0}$ under $\vec{\boldsymbol{\alpha}}$)

monotone if no $\neg \boldsymbol{p_i}$ among the labels

Any monotone HGP based on $\boldsymbol{H}$ computes a sub-function of $\boldsymbol{f_H}$

✓ HGPs based on hypergraphs of **degree** ≤ 2 are polynomially equivalent to **nondeterministic branching programs** (NBPs)



$\mathbf{HGP}^2 = \mathbf{NBP} = \mathbf{NL/poly}$

functions computable by NLogSpace TMs with polynomial advice functions

(non-uniform analogue of NLogSpace)

Rewritings for OMQs with ontologies of depth 1

$\mathbf{HGP}^2 = \mathbf{NL/poly} \subseteq \mathbf{P/poly}$ (for monotone functions)

$\longrightarrow$ polynomial-size **NDL**-rewritings

there is a monotone f computable by a polynomial-size NBP, but
any monotone Boolean formula computing f is of size $n^{\Omega(\log n)}$

$\longrightarrow$ $\exists$ OMQ with superpolynomial **PE**-rewritings only

all OMQs have polynomial **FO**-rewritings $\longleftrightarrow$ $\mathbf{NC}^1 = \mathbf{NL/poly}$

all OMQs with CQs of bounded treewidth have polynomial PE-rewritings

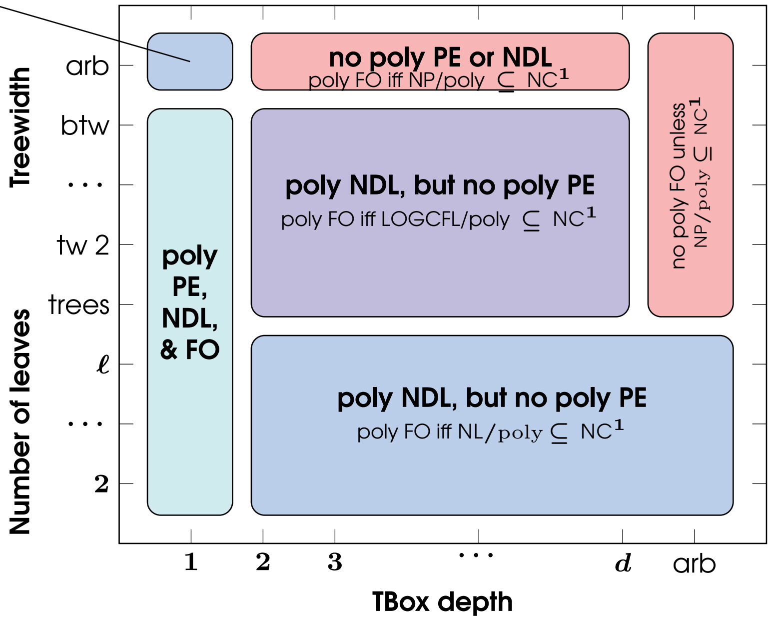
all tree-shaped OMQs have polynomial-size Π_4-rewritings ($\wedge\vee\wedge\vee$)

(SPARQL queries under *OWL 2 QL* entailment regime)

Succinctness landscape for OMQ rewritings

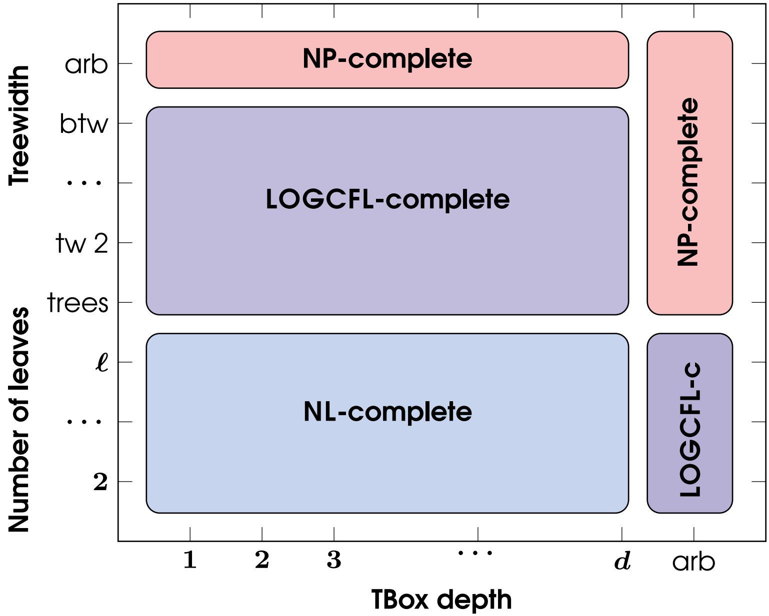
poly NDL, but no poly PE

poly FO iff NL/poly $\subseteq$ NC^1



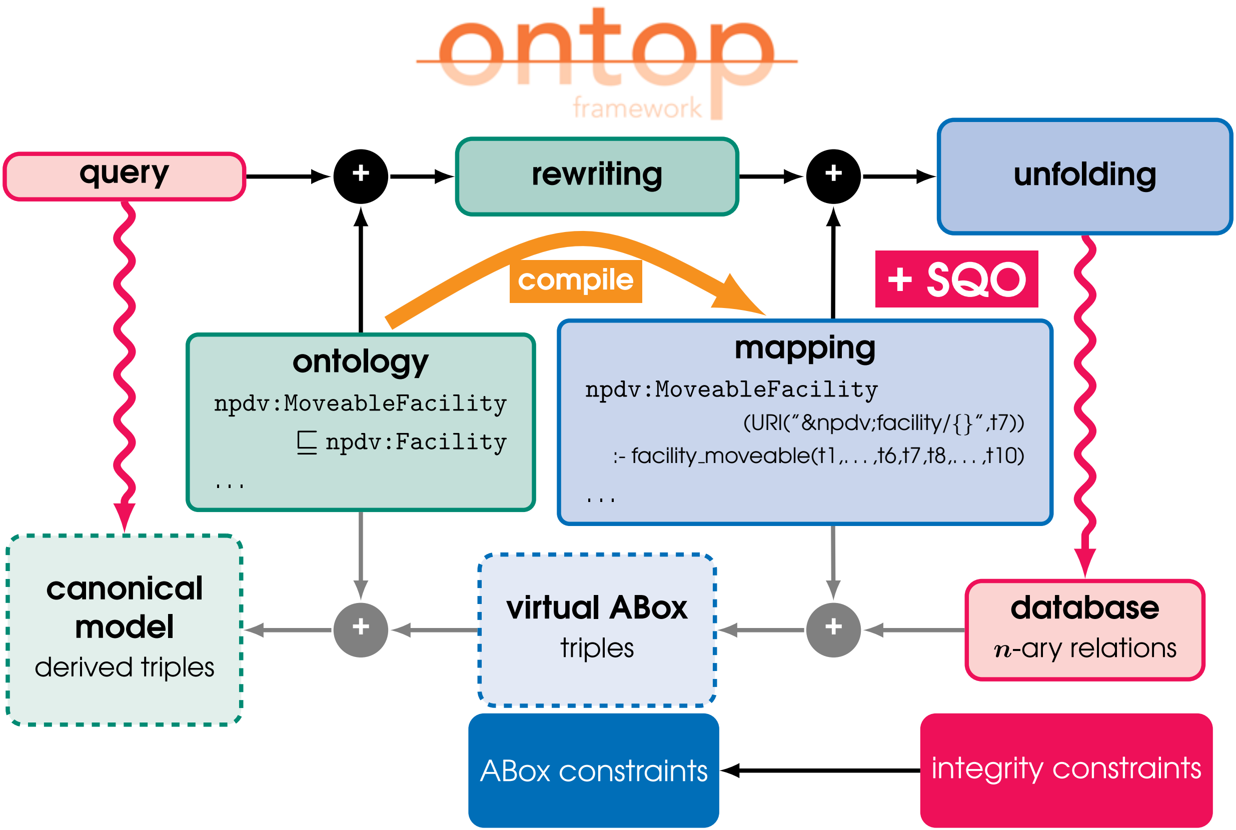
Bienvenu, Kikot, Kontchakov, Podolskii, Z 2012–15

Combined complexity landscape



- CQ evaluation over databases is **NP**-complete **$L \subseteq NL \subseteq LOGCFL \subseteq NC^2 \subseteq P \subseteq NP$**
- bounded treewidth CQ evaluation is **LOGCFL**-complete (logspace reducible to a CFL)

Gottlob et al. 2001



Rodriguez-Muro, Calvanese, Kontchakov, Rezk, Xiao, Z 2010–15

Ontop in practice

✓ T-mappings compile (big parts of) OWL 2 QL ontologies into mappings
(domain and range constraints, concept and role hierarchies)

✓ can be optimised **offline**

✓ few tree witnesses in real-world OBDA → polynomial-size rewritings

✓ database constraints and SQO significantly simplify T-mappings

→ efficient SQL queries over the data

✗ some important conceptual modelling constructs are missing in *OWL 2 QL*
$A \sqsubseteq B \sqcup C$ $\exists R.A \sqsubseteq B$ `owl:sameAs`

? **islands** of OMQ rewritability & succinctness for expressive languages

iTract: Islands of Tractability in Ontology-Based Data Access

EPSRC UK project:

(i) establish a novel, OMQ-centric approach to OBDA aiming to identify **islands of tractable** OMQs in rich ontology and query languages

(ii) develop uniformly efficient OMQ answering techniques for the identified islands

(iii) implement and test these techniques in practice, using state-of-the-art OBDA systems

Team:

– London: MZ (PI), S Kikot (RA), R Kontchakov (co-I), I Razgon (co-I)

– Liverpool: F Wolter (PI), F Papacchini (RA), A Hernich, B Konev

Project partners:

– University of Bozen-Bolzano (Diego Calvanese)
– University of Bremen (Carsten Lutz)
– University of Oslo (Arild Waaler)
– IBM Watson, New York (Mariano Rodriguez-Muro)