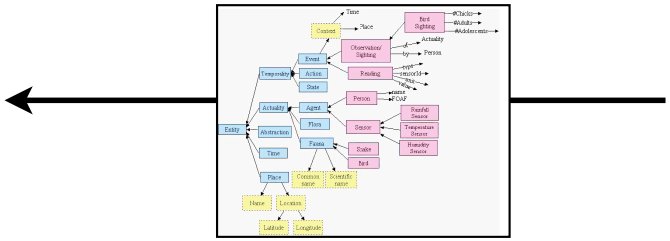
ONTOLOGY-MEDIATED QUERY ANSWERING

Harnessing Knowledge to Get More From Data

Meghyn Bienvenu (*CNRS, University of Montpellier, Inria*)

ONTOLOGY-MEDIATED QUERY ANSWERING (OMQA)

incomplete database

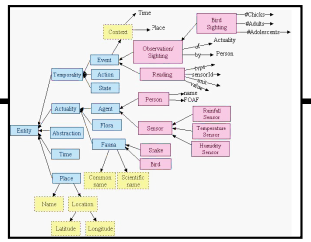


ontology

domain knowledge

user query

ONTOLOGY-MEDIATED QUERY ANSWERING (OMQA)



patient data
"Melanie has listeriosis"
"Paul has Lyme disease"

medical knowledge
"Listeriosis & Lyme disease are bacterial infections"

user query
"Find all patients with bacterial infections"

***expected answers*: Melanie, Paul**

In computer science:

a formal specification of the knowledge of a particular domain, thereby making it amenable to machine processing

Such a specification consists of:

- terminology (or vocabulary) of the domain
- semantic relationships between terms
  - relations of inclusion, equivalence, disjointness, ...

WHAT ARE ONTOLOGIES GOOD FOR?

To **standardize the terminology** of an application domain

- meaning of terms is constrained, so fewer misunderstandings
- by adopting a common vocabulary, easy to share information

To present an **intuitive and unified view of data sources**

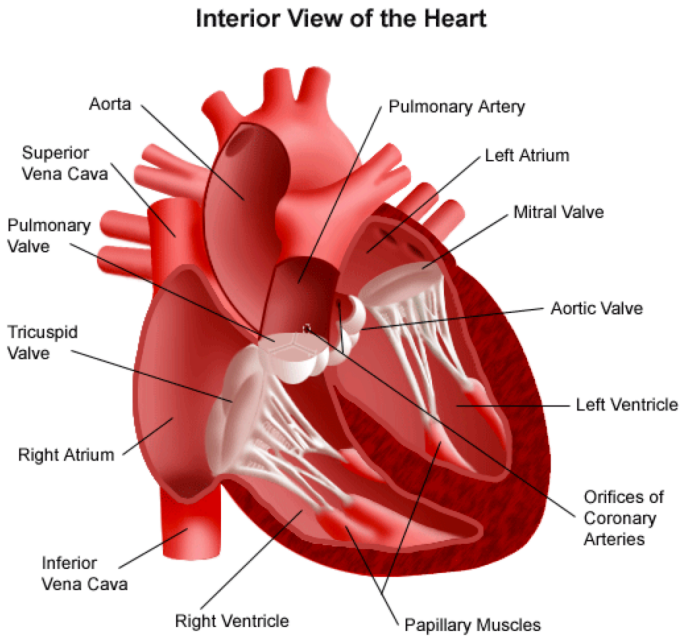
- ontology can be used to **enrich the data vocabulary**, making it easier for users to formulate their queries
- especially useful when **integrating multiple data sources**

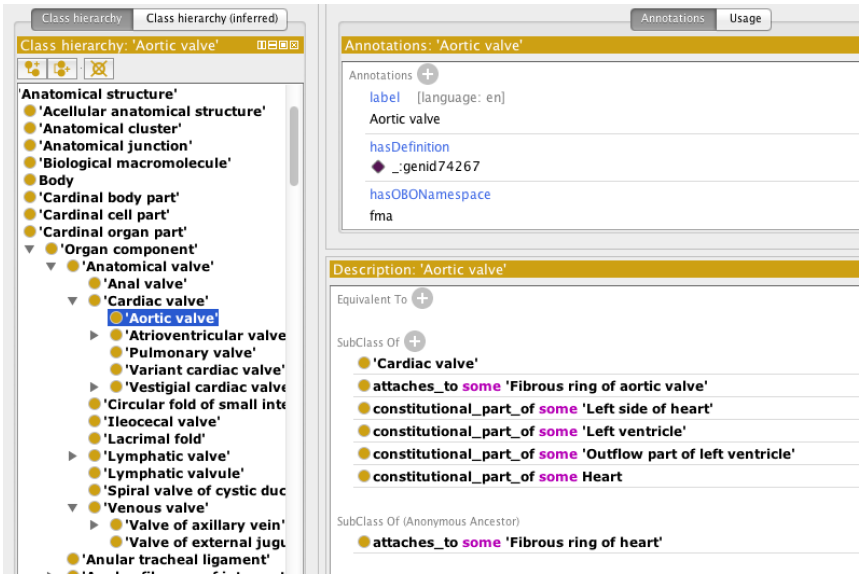
To support **automated reasoning**

- uncover implicit connections between terms, errors in modelling
- **exploit knowledge in the ontology during query answering**, to get back a **more complete set of answers** to queries

APPLICATIONS OF OMQA: MEDICINE

General medical ontologies: SNOMED CT (∼ 300,000 terms!), GALEN
Specialized ontologies: FMA (anatomy), NCI (cancer), ...





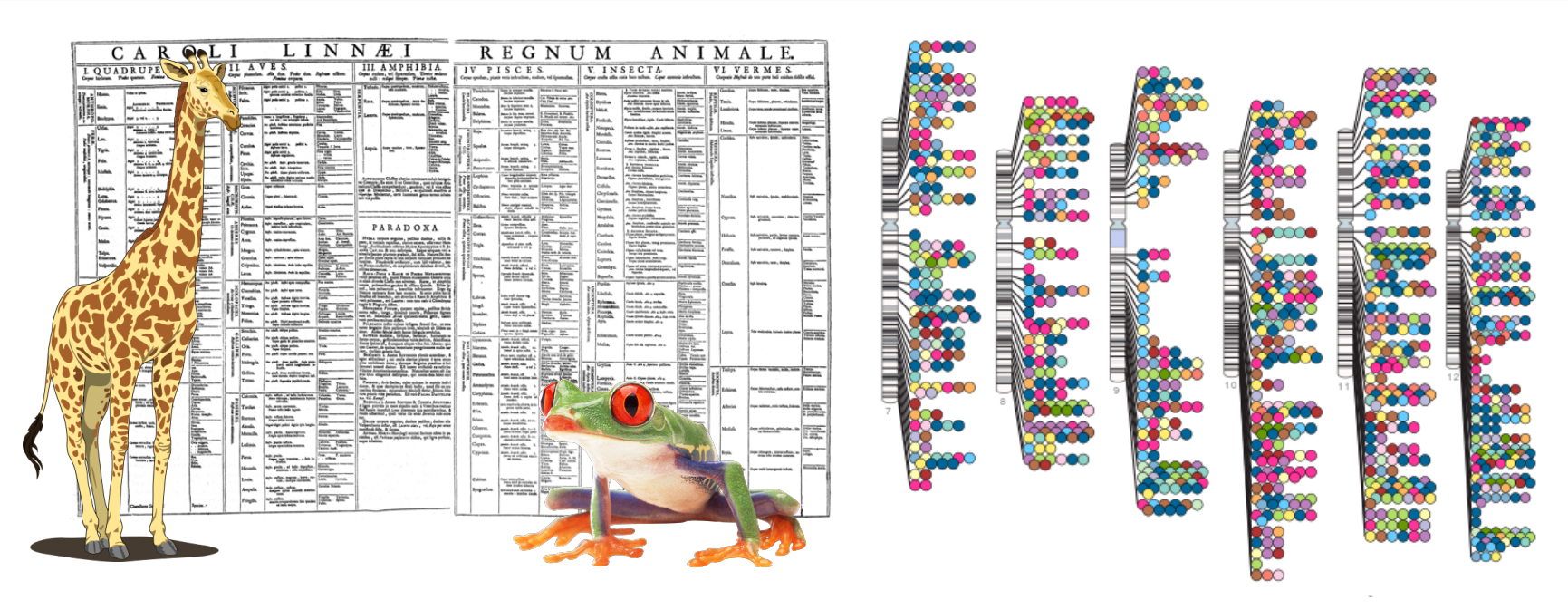
Querying & exchanging medical records (find patients for medical trials)

- myocardial infarction vs. MI vs. heart attack vs. 410.0

Supports tools for annotating and visualizing patient data (scans, x-rays)

APPLICATIONS OF OMQA: LIFE SCIENCES

Hundreds of ontologies at BioPortal (http://bioportal.bioontology.org/):
Gene Ontology (GO), Cell Ontology, Pathway Ontology, Plant Anatomy, ...



Help scientists share, query, & visualize experimental data

APPLICATIONS OF OMQA: ENTREPRISE INFORMATION SYSTEMS

Companies and organizations have lots of data

- need easy and flexible access to support decision-making

Example industrial projects:

- Public debt data: Sapienza Univ. & Italian Department of Treasury
- Energy sector: Optique EU project (several univ, StatOil, Siemens)

OUR FOCUS: DESCRIPTION LOGIC ONTOLOGIES

Description logics (DLs):

- popular means for specifying ontologies
- basis of the web ontology language OWL (W3C standard)

Formally: **decidable fragments of first-order logic**

- inherit well-defined semantics
- succinct, variable-free syntax

OUR FOCUS: DESCRIPTION LOGIC ONTOLOGIES

Description logics (DLs):

- popular means for specifying ontologies
- basis of the web ontology language OWL (W3C standard)

Formally: decidable fragments of first-order logic

- inherit well-defined semantics
- succinct, variable-free syntax

Computational properties well understood (decidability, complexity)

Many implemented reasoners and tools available for use

PLAN FOR THIS TALK

Introduction to DLs & OMQA

Query Rewriting: Limits and Possibilities

Inconsistency Handling in OMQA

INTRODUCTION TO DLS & OMQA

DESCRIPTION LOGIC ONTOLOGIES

Building blocks:

- concept names (unary predicates, classes) — Prof Fellow Course
- role names (binary predicates, properties) — Teaches HeadOf

Constructors to build complex descriptions — $\sqcup, \sqcap, \neg, \forall, \exists, \ldots$

$\text{Faculty} \sqcap \neg \text{Prof}$ $\exists \text{Teaches}.\text{GradCourse}$ Teaches^-

DESCRIPTION LOGIC ONTOLOGIES

Building blocks:

- concept names (unary predicates, classes) $\quad \mathsf{Prof} \quad \mathsf{Fellow} \quad \mathsf{Course}$
- role names (binary predicates, properties) $\quad \mathsf{Teaches} \quad \mathsf{HeadOf}$

Constructors to build complex descriptions $\quad \sqcup, \sqcap, \neg, \forall, \exists, \ldots$

$\mathsf{Faculty} \sqcap \neg \mathsf{Prof} \qquad \exists \mathsf{Teaches}.\mathsf{GradCourse} \qquad \mathsf{Teaches}^{-}$

Ontology = set of axioms

- concept inclusions

$\mathsf{Prof} \sqsubseteq \mathsf{Faculty} \qquad \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \qquad \exists \mathsf{Teaches}.\mathsf{GradCourse} \sqsubseteq \mathsf{Prof}$

- role inclusions

$\mathsf{TaughtBy} \sqsubseteq \mathsf{Teaches}^{-} \qquad \mathsf{HeadOf} \sqsubseteq \mathsf{MemberOf}$

Note: allowed constructors and axioms **depends on chosen DL**

Instance queries (IQs): find instances of a given concept or role

$\mathrm{Faculty}(x)$ $\mathrm{Teaches}(x, y)$

Instance queries (IQs): find instances of a given concept or role

$$\mathrm{Faculty}(x) \qquad \mathrm{Teaches}(x, y)$$

Conjunctive queries (CQs) $\sim$ SPJ queries in SQL, BGPs in SPARQL

conjunctions of atoms, some variables can be existentially quantified

$$\exists y.\, \mathrm{Faculty}(x) \land \mathrm{Teaches}(x, y)$$

(find all faculty members that teach something)

Instance queries (IQs): find instances of a given concept or role

$$\text{Faculty}(x) \qquad \text{Teaches}(x, y)$$

Conjunctive queries (CQs) $\sim$ SPJ queries in SQL, BGPs in SPARQL

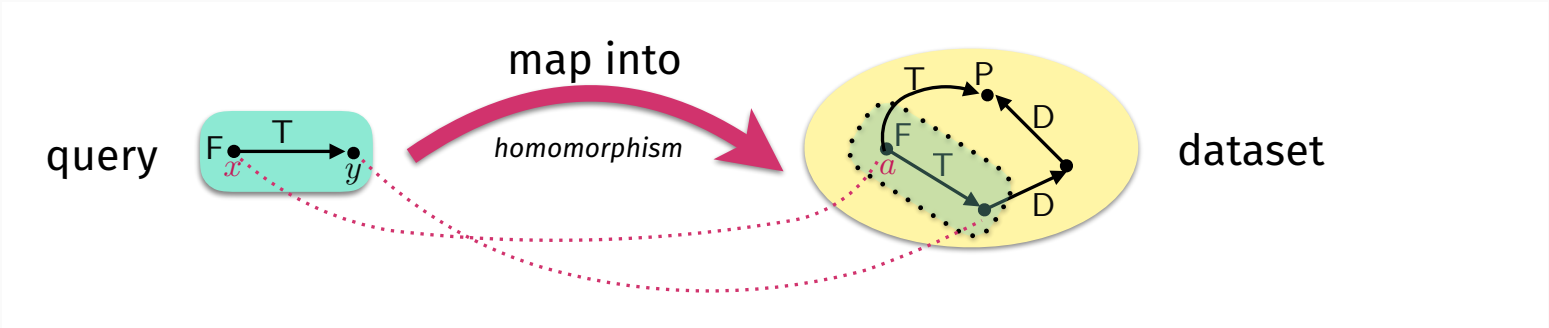
conjunctions of atoms, some variables can be existentially quantified

$$\exists y.\, \text{Faculty}(x) \wedge \text{Teaches}(x, y)$$

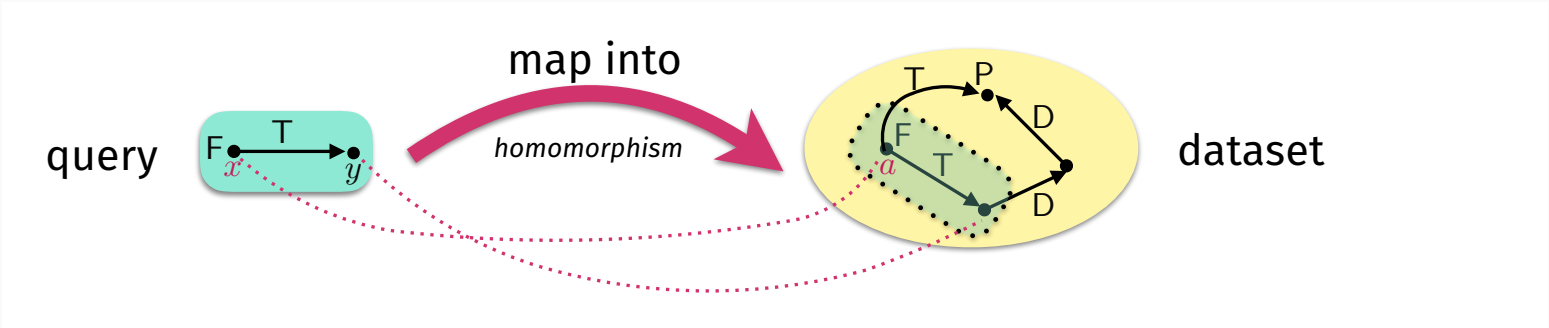
(find all faculty members that teach something)

Unions of conjunctive queries (UCQs): disjunction of CQs

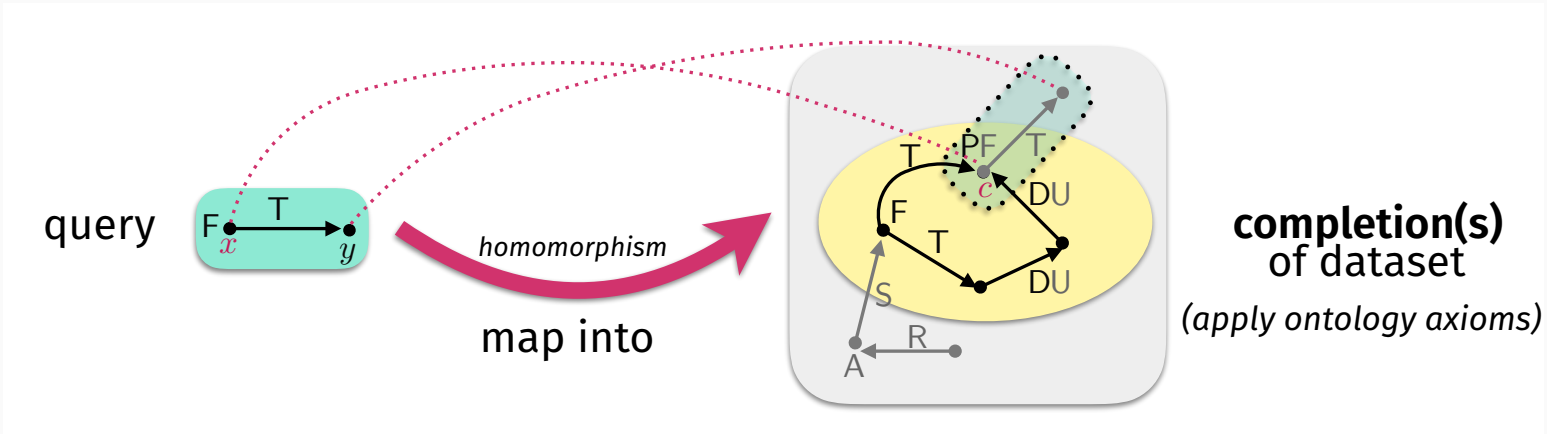
Answering CQs in database setting



Answering CQs in database setting



Answering CQs in the presence of an ontology



EXAMPLE: OMQA WITH DL-LITE ONTOLOGIES

Ontology, expressed in **DL-Lite**: (lightweight DL designed for OMQA)

$\sim$ OWL 2 QL

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

Dataset:

$$\mathcal{D}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

Query: $q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$

EXAMPLE: OMQA WITH DL-LITE ONTOLOGIES

Ontology, expressed in **DL-Lite**: (lightweight DL designed for OMQA)

$\sim$ OWL 2 QL

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

Dataset:

$$\mathcal{D}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

Query: $q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$

Get the following answers:

- anna — $\mathsf{Prof}(\mathsf{anna}) + \mathsf{Prof} \sqsubseteq \mathsf{Faculty} + \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches}$
- tom — $\mathsf{Fellow}(\mathsf{tom}) + \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} + \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})$

QUERY REWRITING: LIMITS AND POSSIBILITIES

FIRST-ORDER QUERY REWRITING

Idea: reduce OMQA to database query evaluation

- rewriting step: ontology $\mathcal{O}$ + query $q \rightsquigarrow$ first-order (FO) query q'

  FO queries $\sim$ SQL queries

- evaluation step: evaluate query q' over dataset

Advantage: harness efficiency of relational database (DB) systems

FIRST-ORDER QUERY REWRITING

Idea: reduce OMQA to database query evaluation

- rewriting step: ontology $\mathcal{O}$ + query $q \rightsquigarrow$ first-order (FO) query q'

  FO queries $\sim$ SQL queries
- evaluation step: evaluate query q' over dataset

Advantage: harness efficiency of relational database (DB) systems

FO-rewriting of q w.r.t. $\mathcal{O}$: FO-query q' such that for every dataset $\mathcal{D}$:

evaluating q' over $\mathcal{D}$ (viewed as DB) gives **correct result**

EXAMPLE: QUERY REWRITING

Same ontology, data, and query as earlier:

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

$$\mathcal{D}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

$$q_1(x) = \exists y.\mathsf{Faculty}(x) \land \mathsf{Teaches}(x, y)$$

EXAMPLE: QUERY REWRITING

Same ontology, data, and query as earlier:

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

$$\mathcal{D}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

$$q_1(x) = \exists y.\mathsf{Faculty}(x) \land \mathsf{Teaches}(x, y)$$

FO-rewriting of $q_1(x)$ w.r.t. $\mathcal{O}$: (disjuncts = different ways to satisfy q_1)

$$(\mathsf{Faculty}(x) \land \exists y.\mathsf{Teaches}(x, y)) \lor (\mathsf{Fellow}(x) \land \exists y.\mathsf{Teaches}(x, y)) \lor \mathsf{Prof}(x)$$

EXAMPLE: QUERY REWRITING

Same ontology, data, and query as earlier:

$$\mathsf{Prof} \sqsubseteq \mathsf{Faculty} \qquad \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} \qquad \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow}$$
$$\mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} \qquad \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} \qquad \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course}$$

$$\mathcal{D}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

$$q_1(x) = \exists y.\mathsf{Faculty}(x) \land \mathsf{Teaches}(x, y)$$

FO-rewriting of $q_1(x)$ w.r.t. $\mathcal{O}$: (disjuncts = different ways to satisfy q_1)

$$(\mathsf{Faculty}(x) \land \exists y.\mathsf{Teaches}(x, y)) \lor (\mathsf{Fellow}(x) \land \exists y.\mathsf{Teaches}(x, y)) \lor \mathsf{Prof}(x)$$

Evaluating the rewriting over $\mathcal{D}_1$ yields:

anna (matches 3rd disjunct) tom (matches 2nd disjunct)

QUERY REWRITING IN DL-LITE

For DL-Lite ontologies, we know FO-rewritings exist for all CQs.

Lots of implemented rewriting algorithms for DL-Lite

- many produce rewritings in the form of UCQs (like in our example)

QUERY REWRITING IN DL-LITE

For DL-Lite ontologies, we know FO-rewritings exist for all CQs.

Lots of implemented rewriting algorithms for DL-Lite

- many produce rewritings in the form of UCQs (like in our example)

Experiments showed that such rewritings can be huge!

- can be difficult / impossible to generate and evaluate

QUERY REWRITING IN DL-LITE

For DL-Lite ontologies, we know FO-rewritings exist for all CQs.

Lots of implemented rewriting algorithms for DL-Lite

- many produce rewritings in the form of UCQs (like in our example)

Experiments showed that such rewritings can be huge!

- can be difficult / impossible to generate and evaluate

Easy to show smallest UCQ-rewriting may be exponentially large:

For DL-Lite ontologies, we know FO-rewritings exist for all CQs.

Lots of implemented rewriting algorithms for DL-Lite

- many produce rewritings in the form of UCQs (like in our example)

Experiments showed that such rewritings can be huge!

- can be difficult / impossible to generate and evaluate

Easy to show smallest UCQ-rewriting may be exponentially large:

Query: $A_1^0(x) \land \ldots \land A_n^0(x)$ Ontology: $A_i^1 \sqsubseteq A_i^0$ $(i = 1, \ldots, n)$

Rewriting: $\bigvee_{(i_1,\ldots,i_n)\in\{0,1\}} A_1^{i_1}(x) \land A_1^{i_1}(x) \land \ldots \land A_1^{i_1}(x)$

For DL-Lite ontologies, we know FO-rewritings exist for all CQs.

Lots of implemented rewriting algorithms for DL-Lite

- many produce rewritings in the form of UCQs (like in our example)

Experiments showed that such rewritings can be huge!

- can be difficult / impossible to generate and evaluate

Easy to show smallest UCQ-rewriting may be exponentially large:

Query: $A_1^0(x) \wedge \ldots \wedge A_n^0(x)$ Ontology: $A_i^1 \sqsubseteq A_i^0$ $(i = 1, \ldots, n)$

Rewriting: $\bigvee_{(i_1,\ldots,i_n)\in\{0,1\}} A_1^{i_1}(x) \wedge A_1^{i_1}(x) \wedge \ldots \wedge A_1^{i_1}(x)$

But equivalent to polysize query $\bigwedge_{i=1}^{n}(A_i^0(x) \vee A_i^1(x))$ (DNF vs CNF)

SIZE OF REWRITINGS IN DL-LITE

Different shapes of rewritings:

- UCQs — disjunction of conjunctions
- positive existential (PE) queries — can mix $\vee$ and $\wedge$
- non-recursive datalog (NDL) queries — allows structure sharing
- first-order (FO) queries — can also use $\neg, \forall$

SIZE OF REWRITINGS IN DL-LITE

Different shapes of rewritings:

- UCQs — disjunction of conjunctions
- positive existential (PE) queries — can mix $\vee$ and $\wedge$
- non-recursive datalog (NDL) queries — allows structure sharing
- first-order (FO) queries — can also use $\neg, \forall$

Question: Do we always have polysize (PE / NDL / FO) rewriting?

SIZE OF REWRITINGS IN DL-LITE

Different shapes of rewritings:

- UCQs — disjunction of conjunctions
- positive existential (PE) queries — can mix $\vee$ and $\wedge$
- non-recursive datalog (NDL) queries — allows structure sharing
- first-order (FO) queries — can also use $\neg, \forall$

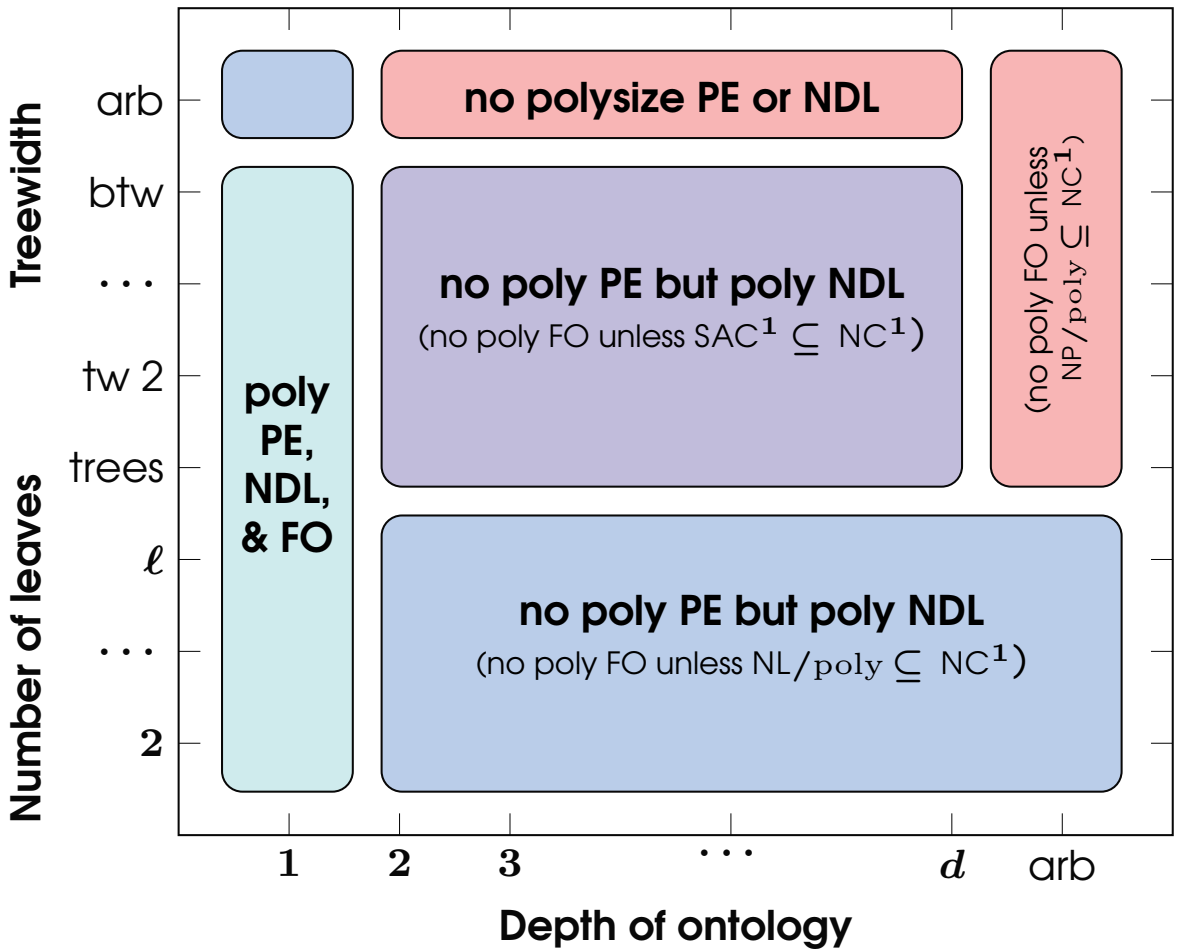
Question: ~~Do we always have polysize (PE / NDL / FO) rewriting?~~ no

When are polysize (PE/ NDL / FO) rewritings possible?

Natural restrictions:

- ontology: bound existential depth
- query: bound treewidth / number of leaves (tree-shaped queries)

SUCCINCTNESS LANDSCAPE



Bad news:

almost never have polysize PE-rewritings

Good news:

polysize NDL-rewritings for many cases

Even better:

NDL-rewritings with optimal complexity

(NL or LOGCFL)

Key technique: link rewriting size to circuit complexity

BEYOND DL-LITE: EXISTENCE OF FO-REWRITINGS

For most other ontology languages: **FO-rewritings may not exist**!

Take for example the **lightweight DL $\mathcal{EL}$** basis for **OWL 2 EL**

no FO-rewriting of $A(x)$ w.r.t. $\mathcal{O} = \{\exists R.A \sqsubseteq A\}$

BEYOND DL-LITE: EXISTENCE OF FO-REWRITINGS

For most other ontology languages: **FO-rewritings may not exist**!

Take for example the **lightweight DL $\mathcal{EL}$** basis for **OWL 2 EL**

no FO-rewriting of $A(x)$ w.r.t. $\mathcal{O} = \{\exists R.A \sqsubseteq A\}$

Hope: FO-rewritings do exist for **typical queries and ontologies**

Question: how to **identify these good cases**? (extend applicability)

- does query q have an FO-rewriting w.r.t. ontology $\mathcal{O}$?

BEYOND DL-LITE: EXISTENCE OF FO-REWRITINGS

For most other ontology languages: **FO-rewritings may not exist**!

Take for example the **lightweight DL $\mathcal{EL}$** basis for **OWL 2 EL**

no FO-rewriting of $A(x)$ w.r.t. $\mathcal{O} = \{\exists R.A \sqsubseteq A\}$

Hope: FO-rewritings do exist for **typical queries and ontologies**

Question: how to **identify these good cases**? (extend applicability)

- does query q have an FO-rewriting w.r.t. ontology $\mathcal{O}$?

Tackled in **series of recent papers** (incl: IJCAI'16 paper)

- decision procedures for **variety of DLs, query languages**
- practical methods for **building rewritings** promising first results

INCONSISTENCY HANDLING IN OMQA

HANDLING DATA INCONSISTENCIES

In realistic settings, can expect some errors in the data

- data likely to be inconsistent with the ontology

Standard semantics: everything is implied - not informative!

HANDLING DATA INCONSISTENCIES

In realistic settings, can expect some errors in the data

- data likely to be inconsistent with the ontology

Standard semantics: everything is implied - not informative!

Two approaches to inconsistency handling:

- resolve the inconsistencies
  - preferable, but not always applicable!
- live with the inconsistencies - adopt alternative semantics
  - meaningful answers to queries despite inconsistencies

PLAUSIBLE ANSWERS: AR SEMANTICS

Repair: $\subseteq$-maximal subset of the data consistent with the ontology

- ways to achieve consistency, keeping as much information as possible

PLAUSIBLE ANSWERS: AR SEMANTICS

Repair: $\subseteq$-maximal subset of the data consistent with the ontology

- ways to achieve consistency, keeping as much information as possible

Plausible answers: hold no matter which repair is chosen

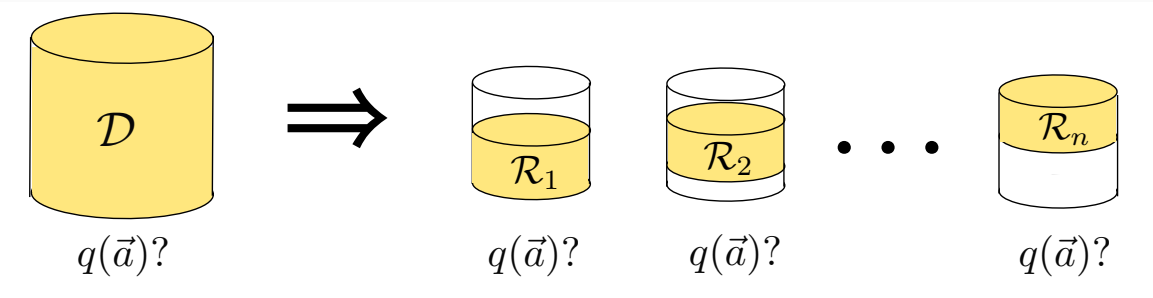
PLAUSIBLE ANSWERS: AR SEMANTICS

Repair: $\subseteq$-maximal subset of the data consistent with the ontology

- ways to achieve consistency, keeping as much information as possible

Plausible answers: hold no matter which repair is chosen

AR semantics: query each repair separately, intersect results



EXAMPLE: AR SEMANTICS

Same ontology and query as before, **add** Prof(tom) to dataset:

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg\mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists\mathsf{Teaches} & \exists\mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists\mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

$$\mathcal{D}_2 = \{\mathsf{Prof}(\mathsf{anna}), \underline{\mathsf{Fellow}(\mathsf{tom})}, \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101}), \underline{\mathsf{Prof}(\mathsf{tom})}\}$$

$$q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$$

EXAMPLE: AR SEMANTICS

Same ontology and query as before, **add** Prof(tom) to dataset:

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

$$\mathcal{D}_2 = \{\mathsf{Prof}(\mathsf{anna}), \underline{\mathsf{Fellow}(\mathsf{tom})}, \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101}), \underline{\mathsf{Prof}(\mathsf{tom})}\}$$

$$q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$$

Two repairs of $\mathcal{D}_2$ w.r.t. $\mathcal{O}$:

$\mathcal{R}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$ drop Prof(tom)

$\mathcal{R}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Prof}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$ drop Fellow(tom)

EXAMPLE: AR SEMANTICS

Same ontology and query as before, **add** Prof(tom) to dataset:

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

$$\mathcal{D}_2 = \{\mathsf{Prof}(\mathsf{anna}), \underline{\mathsf{Fellow}(\mathsf{tom})}, \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101}), \underline{\mathsf{Prof}(\mathsf{tom})}\}$$

$$q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$$

Two repairs of $\mathcal{D}_2$ w.r.t. $\mathcal{O}$:

$\mathcal{R}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$ drop Prof(tom)

$\mathcal{R}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Prof}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$ drop Fellow(tom)

Under AR semantics:

- anna and tom are both answers to q_1
- cs101 is not an answer

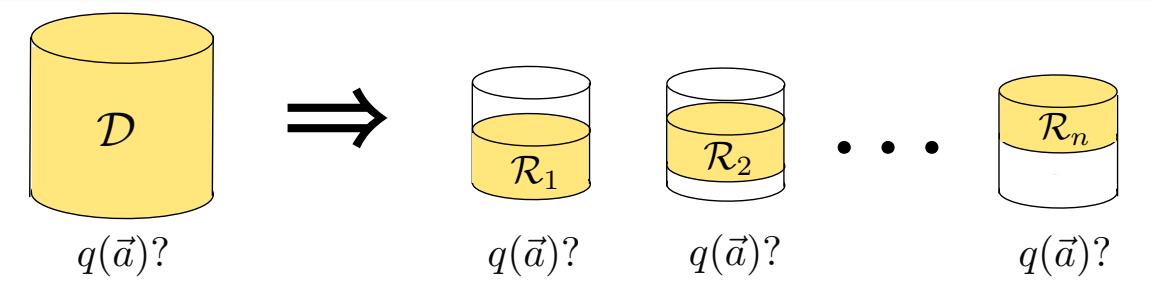
PLAUSIBLE ANSWERS: AR SEMANTICS

Repair: $\subseteq$-maximal subset of the data consistent with the ontology

- ways to achieve consistency, keeping as much information as possible

Plausible answers: hold no matter which repair is chosen

AR semantics: query each repair separately, intersect results



Bad news: query answering under AR semantics is **intractable**

(**coNP-hard** in the size of the data)

Worse: intractable **even in very restricted settings** ($\mathcal{O} = \{A \sqsubseteq \neg B\}$)

TRACTABILITY THROUGH APPROXIMATION

Idea: approximate AR semantics from above and below

TRACTABILITY THROUGH APPROXIMATION

Idea: approximate AR semantics from above and below

Brave semantics **possible answers**

- answer required to hold w.r.t. at least one repair

TRACTABILITY THROUGH APPROXIMATION

Idea: approximate AR semantics from above and below

Brave semantics — possible answers

- answer required to hold w.r.t. at least one repair

IAR semantics — surest answers

- query the intersection of all repairs (surest facts)

TRACTABILITY THROUGH APPROXIMATION

Idea: approximate AR semantics from above and below

Brave semantics — possible answers

- answer required to hold w.r.t. at least one repair

IAR semantics — surest answers

- query the intersection of all repairs (surest facts)

Relationship between the semantics:

IAR answers $\subseteq$ AR answers $\subseteq$ brave answers

TRACTABILITY THROUGH APPROXIMATION

Idea: approximate AR semantics from above and below

Brave semantics — possible answers

- answer required to hold w.r.t. at least one repair

IAR semantics — surest answers

- query the intersection of all repairs (surest facts)

Relationship between the semantics:

IAR answers $\subseteq$ AR answers $\subseteq$ brave answers

Good news: these semantics are tractable for DL-Lite ontologies

BACK TO THE EXAMPLE

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

$\mathcal{D}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101}), \mathsf{Prof}(\mathsf{tom})\}$

$q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$

Repairs of $\mathcal{D}_2$ w.r.t. $\mathcal{O}$:

$$\mathcal{R}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

$$\mathcal{R}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Prof}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

Intersection of repairs: $\{\mathsf{Prof}(\mathsf{anna}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$

BACK TO THE EXAMPLE

$$\mathsf{Prof} \sqsubseteq \mathsf{Faculty} \qquad \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} \qquad \mathsf{Prof} \sqsubseteq \neg\mathsf{Fellow}$$
$$\mathsf{Prof} \sqsubseteq \exists\mathsf{Teaches} \qquad \exists\mathsf{Teaches} \sqsubseteq \mathsf{Faculty} \qquad \exists\mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course}$$

$\mathcal{D}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101}), \mathsf{Prof}(\mathsf{tom})\}$

$q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$

Repairs of $\mathcal{D}_2$ w.r.t. $\mathcal{O}$:

$$\mathcal{R}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$
$$\mathcal{R}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Prof}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

Intersection of repairs: $\{\mathsf{Prof}(\mathsf{anna}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$

Brave answers: anna tom

BACK TO THE EXAMPLE

$$\begin{array}{lll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} \\ \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course} \end{array}$$

$\mathcal{D}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101}), \mathsf{Prof}(\mathsf{tom})\}$

$q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$

Repairs of $\mathcal{D}_2$ w.r.t. $\mathcal{O}$:

$\mathcal{R}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$

$\mathcal{R}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Prof}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$

Intersection of repairs: $\{\mathsf{Prof}(\mathsf{anna}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$

Brave answers: anna tom IAR answers: anna

CQAPri first system for AR query answering in DL-Lite

Implements hybrid approach:

- compute IAR and brave answers — polytime
  - gives upper and lower bounds on AR answers
- use SAT solvers to identify remaining AR answers
- three categories of answers: possible, likely, (almost) sure

Interaction with user:

- explaining query results
  - why a possible answer? why not a sure answer?
- query-driven repairing (IJCAI'16)
  - exploit user feedback to improve data quality

CONCLUSION & OUTLOOK

Promising approach, important applications

Significant recent advances, but still lots left to do!

- efficiency: more expressive ontology and query languages
- robustness: inconsistencies, uncertainty, vagueness, exceptions
- usability: ontology construction, explanation, query formulation, ...

Connections to other disciplines: databases, Semantic Web, theory

Collaboration with other AI areas?

- natural language processing
- machine learning

Questions ?

See paper for references & pointers to the literature

Based on joint work with:
Camille Bourgaux, Balder ten Cate, François Goasdoué, Peter Hansen, Carsten Lutz, Stanislav Kikot, Roman Kontchakov, Magdalena Ortiz, Vladimir Podolskii, Riccardo Rosati, Mantas Šimkus, Frank Wolter, Guohui Xiao, Michael Zakharyschev