QUERY ANSWERING WITH DESCRIPTION LOGIC ONTOLOGIES

Meghyn Bienvenu (*CNRS & Université de Montpellier*)
Magdalena Ortiz (*Vienna University of Technology*)

RESEARCH TRENDS IN OMQA

Lots of work on developing and implementing efficient OMQA algorithms

Focus mostly on DL-Lite (and related dialects):

- First algorithm PerfectRef proposed in mid-2000's
- Rewrites into UCQs, implemented in QUONTO
- Improved versions proposed in REQUIEM, PRESTO, RAPID, …
- Some algorithms rewrite into positive existential queries or Datalog programs instead of UCQs
- Resulting queries are smaller, can be easier to evaluate

Tractable classes, fragments of lower complexity

Rewriting engines for other Horn DLs also developed, e.g.,

- Requiem and the related Kyrie cover several $\mathcal{EL}$ dialects
- Clipper, and recently Rapid cover Horn-$\mathcal{SHIQ}$

They usually rewrite into Datalog programs

Much attention devoted to understanding the limits of rewritability and size of rewritings

When are polynomial rewritings possible?

Can we give bounds on the size of rewritings?

Which non-DL-Lite ontologies can be rewritten into FO-queries?

$\rightsquigarrow$ related to non-uniform complexity:

- study specific pairs $(q, \mathcal{T})$, called ontology-mediated queries

Saturate the ABox using the TBox axioms

$\rightsquigarrow$ a finite version of the canonical model

and then evaluate the query over the saturated ABox

Two approaches:

- modify the query before evaluation to ensure soundness, or
- evaluate and then filter unsound answers

First proposed for $\mathcal{EL}$, then also for DL-Lite

Extended to other dialects, richer DLs

This course: assume data given as ABox assertions (unary + binary)

Problem: how to query existing relational data (arbitrary arity)?

Solution: use mapping that specifies relationship between the database relations and the concepts / roles in DL vocabulary

Formally: mapping assertions of the form $\varphi \rightarrow \psi$ where:

- φ is an query formulated using DB relations
- ψ is a query in the DL vocabulary

Global-as-view (GAV) mappings: φ CQ, ψ atom (no quantifiers)

Handling mappings:

- apply mappings to generate ABox, proceed as usual
- virtual ABox: unfolding step to get rewriting over DB relations

OTHER RESEARCH TOPICS (NON-EXHAUSTIVE)

Beyond classical OMQA

- inconsistency-tolerant query answering
- probabilistic query answering
- privacy-aware query answering
- temporal query answering

Support for building and maintaining OMQA systems

- module extraction
- ontology evolution
- query inseparability and emptiness

Improving the usability of OMQA systems

- interfaces and support for query formulation
- explaining query (non-)answers
- combining complete and incomplete information

ZOOM: INCONSISTENCY-TOLERANT QUERY ANSWERING

HANDLING DATA INCONSISTENCIES

In realistic settings, can expect some errors in the data

- ABox likely to be inconsistent with the TBox

Standard semantics: all tuples are returned - not informative!

In realistic settings, can expect some errors in the data

- ABox likely to be inconsistent with the TBox

Standard semantics: all tuples are returned - 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

EXAMPLE: WHICH ANSWERS TO RETURN?

Consider the following TBox $\mathcal{T}$:

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

the ABox

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

and the query $q(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$

Consider the following TBox $\mathcal{T}$:

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

the ABox

$$\mathcal{A} = \{\text{Prof(anna)}, \underline{\text{Fellow(tom)}}, \text{Teaches(tom, cs101)}, \underline{\text{Prof(tom)}}\}$$

and the query $q(x) = \exists y.\text{Faculty}(x) \wedge \text{Teaches}(x, y)$

Which individuals should be returned (or not returned as answers)?

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

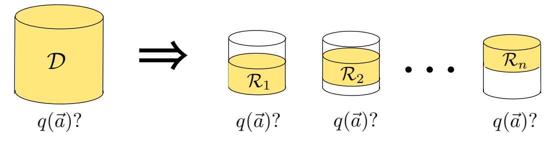
Plausible answers: hold no matter which repair is chosen

Repair: ⊆-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

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

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

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

EXAMPLE: AR SEMANTICS

$\mathsf{Prof} \sqsubseteq \mathsf{Faculty} \qquad \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} \qquad \mathsf{Prof} \sqsubseteq \neg\mathsf{Fellow}$

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

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

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

Two repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$:

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

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

EXAMPLE: AR SEMANTICS

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

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

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

Two repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$:

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

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

Under AR semantics:

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

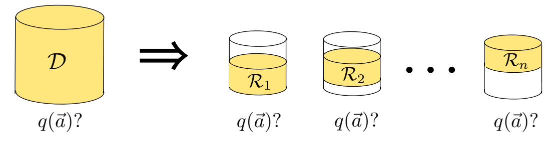
PLAUSIBLE ANSWERS: AR SEMANTICS

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 ($\mathcal{T} = \{A \sqsubseteq \neg B\}$)

Idea: approximate AR semantics from above and below

Idea: approximate AR semantics from above and below

Brave semantics **possible answers**

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

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)

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

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

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

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

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

Repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$:

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

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

BACK TO THE EXAMPLE

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

$$\mathcal{A} = \{\text{Prof}(\text{anna}), \text{Fellow}(\text{tom}), \text{Teaches}(\text{tom}, \text{cs101}), \text{Prof}(\text{tom})\}$$

$$q(x) = \exists y.\text{Faculty}(x) \land \text{Teaches}(x, y)$$

Repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$:

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

Brave answers:

IAR answers:

BACK TO THE EXAMPLE

$\mathsf{Prof} \sqsubseteq \mathsf{Faculty}$ $\mathsf{Fellow} \sqsubseteq \mathsf{Faculty}$ $\mathsf{Prof} \sqsubseteq \neg\mathsf{Fellow}$

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

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

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

Repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$:

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

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

Brave answers: anna tom IAR answers:

BACK TO THE EXAMPLE

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

$$\mathcal{A} = \{\text{Prof}(\text{anna}), \text{Fellow}(\text{tom}), \text{Teaches}(\text{tom}, \text{cs101}), \text{Prof}(\text{tom})\}$$

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

Repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$:

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

Brave answers: anna tom

IAR answers: anna

TOWARDS PRACTICAL SYSTEMS FOR INCONSISTENCY HANDLING

CQAPri first system for AR query answering in DL-Lite

Implements hybrid approach:

- compute IAR and brave answers polytime (data)
  - 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
  - exploit user feedback to improve data quality

ZOOM: COMBINING COMPLETE AND INCOMPLETE INFORMATION

We have seen the classical certain answer semantics:

$$\begin{aligned}\mathcal{T} = \{&\text{BScStud} \sqsubseteq \text{Student}\\ &\text{Student} \sqsubseteq \exists\text{attends.Course}\\ &\text{BScStud} \sqsubseteq \forall\text{attends.}\neg\text{GradCourse}\}\\ \text{q} = {}&\text{attends}(x, y)\end{aligned}$$

We have seen the classical certain answer semantics:

$$\begin{aligned}\mathcal{T} = \{&\mathsf{BScStud} \sqsubseteq \mathsf{Student}\\ &\mathsf{Student} \sqsubseteq \exists \mathsf{attends}.\mathsf{Course}\\ &\mathsf{BScStud} \sqsubseteq \forall \mathsf{attends}.\neg \mathsf{GradCourse}\}\\ \mathsf{q} = {}&\mathsf{attends}(x, y)\end{aligned}$$

Consider the following ABox:

$$\begin{aligned}\mathcal{A} = \{&\mathsf{BScStud}(Ann)\\ &\mathsf{Course}(c_1)\\ &\mathsf{Course}(c_2)\\ &\mathsf{GradCourse}(c_2)\}\end{aligned}$$

COMPLETE VS. INCOMPLETE INFORMATION – USUAL SETTING

We have seen the classical certain answer semantics:

$$\mathcal{T} = \{\text{BScStud} \sqsubseteq \text{Student}$$
$$\text{Student} \sqsubseteq \exists\text{attends.Course}$$
$$\text{BScStud} \sqsubseteq \forall\text{attends.}\neg\text{GradCourse}\}$$
$$\text{q} = \text{attends}(x, y)$$

Consider the following ABox:

$$\mathcal{A} = \{\text{BScStud}(Ann)$$
$$\text{Course}(c_1)$$
$$\text{Course}(c_2)$$
$$\text{GradCourse}(c_2)\}$$

Query Answer: $\emptyset$

Suppose we have complete knowledge about existing courses:

$$\mathcal{T} = \{\text{BScStud} \sqsubseteq \text{Student}$$
$$\text{Student} \sqsubseteq \exists\text{attends.Course}$$
$$\text{BScStud} \sqsubseteq \forall\text{attends.}\neg\text{GradCourse}\}$$
$$q = \text{attends}(x, y)$$
$$\Sigma = \{\text{Course}\}$$

Suppose we have complete knowledge about existing courses:

$$\mathcal{T} = \{\text{BScStud} \sqsubseteq \text{Student}$$
$$\text{Student} \sqsubseteq \exists\text{attends.Course}$$
$$\text{BScStud} \sqsubseteq \forall\text{attends.}\neg\text{GradCourse}\}$$
$$q = \text{attends}(x, y)$$
$$\Sigma = \{\text{Course}\}$$

Recall the previous ABox:

$$\mathcal{A} = \{\text{BScStud}(Ann)$$
$$\text{Course}(c_1)$$
$$\text{Course}(c_2)$$
$$\text{GradCourse}(c_2)\}$$

Suppose we have complete knowledge about existing courses:

$$\begin{aligned}\mathcal{T} = \{&\mathsf{BScStud} \sqsubseteq \mathsf{Student}\\ &\mathsf{Student} \sqsubseteq \exists \mathsf{attends}.\mathsf{Course}\\ &\mathsf{BScStud} \sqsubseteq \forall \mathsf{attends}.\neg \mathsf{GradCourse}\}\\ q = {}& \mathsf{attends}(x, y)\\ \mathbf{\Sigma} = {}& \{\mathsf{Course}\}\end{aligned}$$

Recall the previous ABox:

$$\begin{aligned}\mathcal{A} = \{&\mathsf{BScStud}(Ann)\\ &\mathsf{Course}(c_1)\\ &\mathsf{Course}(c_2)\\ &\mathsf{GradCourse}(c_2)\}\end{aligned}$$

Query Answer: $\{(Ann, c_1)\}$

DLs adopt an open-world view

- Standard FOL-semantics
- expresses incomplete knowledge, many models

A partial closed-world view is desirable

- use partial completeness to infer more answers
- meaningful when data comes from complete DB tables

Closed Predicates in DLs

- We can enrich a KB with a set Σ of concept/role names
- We assume those predicates are complete
- Models of $(\mathcal{T}, \mathcal{A}, \Sigma)$ are models of $(\mathcal{T}, \mathcal{A})$
- Additionally, the extensions of closed predicates must be exactly as given in $\mathcal{A}$

Observation: we need reasoning by cases

- Even for instance checking in lightweight DLs

COMPLEXITY OF REASONING WITH CLOSED PREDICATES

Observation: we need reasoning by cases

- Even for instance checking in lightweight DLs

Closed predicates make reasoning harder

COMPLEXITY OF REASONING WITH CLOSED PREDICATES

Observation: we need reasoning by cases

- Even for instance checking in lightweight DLs

Closed predicates make reasoning harder

- but how much harder?

COMPLEXITY OF REASONING WITH CLOSED PREDICATES

Observation: we need reasoning by cases

- Even for instance checking in lightweight DLs

Closed predicates make reasoning harder

- but how much harder?

What do we know?

- Expressible if the DL is expressive enough
- We can use a construct called nominals

$$\text{Course} \sqsubseteq \{c_1\} \sqcup \{c_2\}$$

We need a DL called $\mathcal{ALCO}$

EXPRESSING CLOSED PREDICATES WITH NOMINALS (CONT.)

We need a DL called $\mathcal{ALCO}$

$$\mathcal{L} \ + \ \text{closed predicates} \subseteq \mathcal{L} \cup \mathcal{ALCO}$$

We need a DL called $\mathcal{ALCO}$

$$\mathcal{L} + \text{closed predicates} \subseteq \mathcal{L} \cup \mathcal{ALCO}$$

This allows us to infer upper bounds in combined complexity

EXPRESSING CLOSED PREDICATES WITH NOMINALS (CONT.)

We need a DL called $\mathcal{ALCO}$

$$\mathcal{L} + \text{ closed predicates} \subseteq \mathcal{L} \cup \mathcal{ALCO}$$

This allows us to infer upper bounds in combined complexity but they are not so good:

- Standard reasoning: consistency, fact entailment in EXP
- Conjunctive query answering: in 2EXP

EXPRESSING CLOSED PREDICATES WITH NOMINALS (CONT.)

We need a DL called $\mathcal{ALCO}$

$$\mathcal{L} + \text{closed predicates} \subseteq \mathcal{L} \cup \mathcal{ALCO}$$

This allows us to infer upper bounds in combined complexity but they are not so good:

- Standard reasoning: consistency, fact entailment in EXP
- Conjunctive query answering: in 2EXP

Optimal?

COMPLEXITY OF STANDARD REASONING

	Without closed predicates	With closed predicates
DL-Lite	NLOGSPACE	NP
DL-Lite$_\mathcal{R}$	NLOGSPACE	NP
$\mathcal{EL}$	P	EXP
$\mathcal{ALCO}$	EXP	EXP
$\mathcal{SHOQ}$, $\mathcal{SHOI}$	EXP	EXP

all are completeness results

	Without closed predicates	With closed predicates
DL-Lite	NP	coNExp-hard
DL-Lite$_{\mathcal{R}}$	NP	**2Exp**
$\mathcal{EL}$	NP	**2Exp**
$\mathcal{ALCO}$	**2Exp**	2Exp
$\mathcal{SHOQ}$, $\mathcal{SHOI}$	2Exp	2Exp

all but red are completeness results

$(\mathsf{q}, \mathcal{T}, \Sigma)$ a **query** q,
a **TBox** $\mathcal{T}$,
set of **closed predicates** Σ

Viewed as a query language, it is **non-monotonic**

$(\mathsf{q}, \mathcal{T}, \Sigma)$ a **query** q,
a TBox $\mathcal{T}$,
set of closed predicates Σ

Viewed as a query language, it is non-monotonic

For OMQs with closed predicates, there are
polynomial rewritings into
disjunctive Datalog with stratified negation

Questions ?