Counting Query Answers over a DL-Lite Knowledge Base

KRDB Summer Seminars
Bozen-Bolzano, Italy

- Joint work with: Diego Calvanese[1,2], Julien Corman[1] and Simon Razniewski[3]
- Speaker: Davide Lanti[1] lanti@inf.unibz.it
- Affiliations:
  - [1] Free University of Bozen-Bolzano, Italy
  - [2] Umeå University, Sweden
  - [3] Max-Planck-Institut für Informatik, Germany

Outline

- The Setting
- Tractability and Intractability
- Rewritability and Non-rewritability
- Conclusions and Future Directions

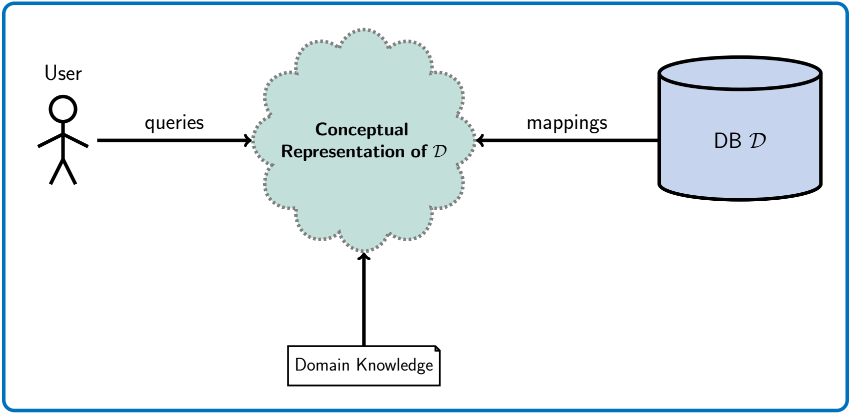
Outline

The Problem of Accessing Relevant Data

- Every day, a huge amount of data is produced by various actors
- Such data is valuable, but it must be accessed and processed to create value
- Complex organization of how the data is stored[1], proper of big companies or institutions, is recognized as one of the huge challenges to data access

[1] E.g., data organized according to complex database schemas involving a significant number of attributes

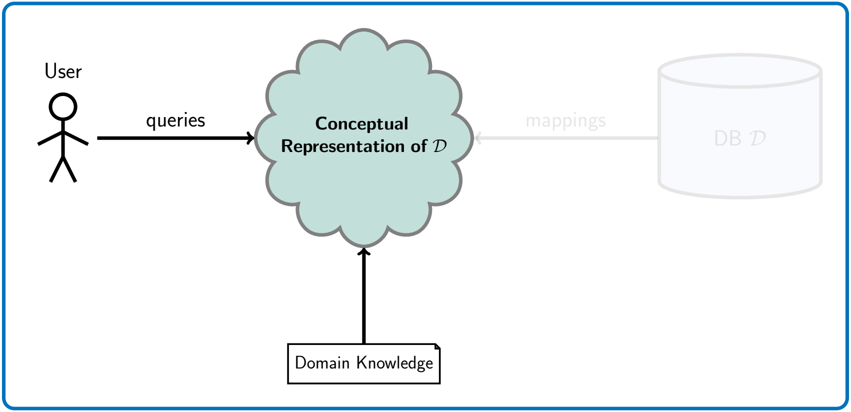
Ontology-based Data Access (OBDA) [Poggi et al., 2008]



OBDA Approach to Data Access

Hide the complexity of data storage behind a convenient representation taking into account both the domain knowledge and the content of the relational database.

Ontology-mediated Query Answering (OMQA) [Bienvenu and Ortiz, 2015]



OMQA, or Query Answering over a Knowledge Base (KB)

We assume the conceptual representation to be materialized, and we ignore mappings and DB.

Syntax

- A Knowledge Base (KB) is a pair $(\mathcal{T}, \mathcal{A})$ where $\mathcal{T}$ is a finite set of axioms called TBox and $\mathcal{A}$ is a finite set of assertions called ABox.
- Axioms in $\mathcal{T}$ are positive inclusions $B \sqsubseteq C$, negative inclusions $B \sqsubseteq \neg C$, and role inclusions $R \sqsubseteq R'$, where concepts B, C and roles R, R' adhere to the following grammar:

$$R \longrightarrow P \mid P^{-} \qquad B \longrightarrow A \mid \geqslant_1 R \qquad C \longrightarrow A \mid \geqslant_n R,$$

  where A is a *concept name*, P is a property name, and $n \in \mathbb{N}^{+}$.
- Assertions in $\mathcal{A}$ are ground atoms of the form $A(a)$, $P(a, b)$, where a, b are constants.
- We distinguish the following fragments of the logic above:
  - *DL-Lite*$_{pos}$ only allows for positive inclusions, with the requirement that $n = 1$.
  - *DL-Lite*$_{core}$ extends *DL-Lite*$_{pos}$ with negative inclusions
  - The superscript $^{\mathcal{H}}$ extends the logic with role inclusions
  - The superscript $^{\mathcal{N}^{-}}$ extends the logic with arbitrary number restrictions $\geqslant n$, but only on the RHS of axioms.

Semantics

- As usual, an interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. Here we assume the standard name assumption:
  - $c^{\mathcal{I}} = c$, for every constant c
- From now on, whenever convenient we treat interpretations as sets of atoms (over constants and anonymous objects)
- $\mathcal{I}$ is a model for a KB $\mathcal{K}$, denoted as $\mathcal{I} \models \mathcal{K}$, if
  - $\mathcal{A} \subseteq \mathcal{I}$
  - $E_1^{\mathcal{I}} \subseteq E_2^{\mathcal{I}}$, for each $E_1 \sqsubseteq E_2 \in \mathcal{T}$

Query Answering under Count Semantics (Definition)

- We use the notation
$$q(\vec{x}) \leftarrow p_1(\vec{t}_1), \ldots, p_n(\vec{t}_n)$$
for conjunctive queries (in particular, the *body* of a query is a set of atoms)
- A match ρ for q in an interpretation $\mathcal{I}$ is a mapping over variables such that $\rho(\mathit{body}(q)) \subseteq \mathcal{I}$
- An answer to $q(\vec{x})$ over $\mathcal{I}$ is a pair (ω, k) such that
  - $k \geqslant 1$
  - there are exactly k matches $\rho_1, \ldots, \rho_k$ for q in $\mathcal{I}$ that verify $\omega = \rho_i|_{\vec{x}}$, for $i \in \{1, \ldots, k\}$
  - We denote by $\mathit{ans}(q, \mathcal{I})$ the set of answers to q over $\mathcal{I}$
  - (ω, k) is a certain answer to q over a KB $\mathcal{K}$, denoted as $(\omega, k) \in \mathit{cert}(q, \mathcal{K})$, if k is the smallest number such that $(\omega, k) \in \mathit{ans}(q, \mathcal{I})$ for some model $\mathcal{I}$ of $\mathcal{K}$.

Query Answering over a KB under Count Semantics (Example)

Knowledge Base

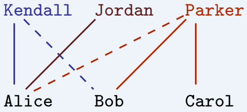
$$\mathcal{A} = \left\{ \begin{array}{l} \mathit{hasChild}(\mathit{Kendall}, \mathit{Alice}), \\ \mathit{hasChild}(\mathit{Jordan}, \mathit{Alice}), \\ \mathit{hasChild}(\mathit{Parker}, \mathit{Bob}), \\ \mathit{hasChild}(\mathit{Parker}, \mathit{Carol}), \\ \mathit{FatherOfTwo}(\mathit{Kendall}), \\ \mathit{FatherOfThree}(\mathit{Parker}) \end{array} \right\}$$

$$\mathcal{T} = \left\{ \begin{array}{l} \mathit{FatherOfTwo} \sqsubseteq\ \geqslant_2 \mathit{hasChild}, \\ \mathit{FatherOfThree} \sqsubseteq\ \geqslant_3 \mathit{hasChild} \\ \exists \mathit{hasChild}^- \sqsubseteq \mathit{Child} \end{array} \right\}$$

Query

$$q() \leftarrow \mathit{Child}(y)$$

Model



Answer: 3

Query Answering over a KB under Count Semantics (Example)

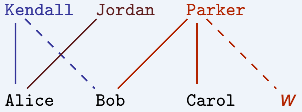
Knowledge Base

$$\mathcal{A} = \left\{ \begin{array}{l} hasChild(Kendall, Alice), \\ hasChild(Jordan, Alice), \\ hasChild(Parker, Bob), \\ hasChild(Parker, Carol), \\ FatherOfTwo(Kendall), \\ FatherOfThree(Parker) \end{array} \right\} \qquad \mathcal{T} = \left\{ \begin{array}{l} FatherOfTwo \sqsubseteq \geqslant_2 hasChild, \\ FatherOfThree \sqsubseteq \geqslant_3 hasChild \\ \exists hasChild^- \sqsubseteq Child \\ Child \sqsubseteq \leqslant_2 hasChild^- \end{array} \right\}$$

Query

$$q() \leftarrow Child(y)$$

Model



Answer: 4

Motivation

- We focus here on the DL-Lite family because it is the language of choice of OBDA/OMQA, specifically designed for rewritability of CQs/UCQs
  - Rewritability is a key notion in OBDA/OMQA, and it guarantees that the certain answers over a knowledge base can be retrieved by just a (rewritten) query over the DB/ABox.
- Counting answers is a basic functionality for a DBMS, and at the basis of analytics tasks
- Number restrictions provide a quantitive measure over incomplete information
  - Can encode statistics about the domain such as population, number of cities, number of accidents, etc.
  - Can be used to identify gaps and inconsistencies in the KB (e.g., retrieve the missing child of Kedall)
  - Can be used to enrich query formulations (e.g., ask for all parents of at least two children)

Outline

- The Setting
- Tractability and Intractability
- Rewritability and Non-rewritability
- Conclusions and Future Directions

Decision Problem

COUNT

Input: *DL-Lite* KB $\mathcal{K}$, boolean CQ q, $k \in \mathbb{N}^+$
Decide: $(\varepsilon, k) \in \mathit{cert}(q, \mathcal{K})$

Data Complexity (Same as [Nikolaou et al., 2019])

We consider as size of the input the size of the ABox, and of k (encoded in binary).

Query Answering under Count Semantics is Hard

Proposition ([Kostylev and Reutter, 2015])

COUNT *is* coNP*-complete for DL-Lite$_{core}$ and CQs.*

- Actually, for this problem we lose two desiderable properties when it comes to tractability:
  - Negative information affects the answers to a query

Query Answering under Count Semantics is Hard

Proposition ([Kostylev and Reutter, 2015])

COUNT *is* CONP*-complete for DL-Lite$_{core}$ and CQs.*

- Actually, for this problem we lose two desiderable properties when it comes to tractability:
  - Negative information affects the answers to a query
  - There is no universal model (see later)

Query Answering under Count Semantics is Hard

Proposition ([Kostylev and Reutter, 2015])

COUNT *is* CONP*-complete for DL-Lite$_{core}$ and CQs.*

- Actually, for this problem we lose two desiderable properties when it comes to tractability:
  - Negative information affects the answers to a query
  - There is no universal model (see later)

Query Answering under Count Semantics is Hard

Proposition ([Kostylev and Reutter, 2015])

COUNT *is* coNP*-complete for DL-Lite*$_{core}$ *and CQs.*

- Actually, for this problem we lose two desiderable properties when it comes to tractability:
  - Negative information affects the answers to a query
  - There is no universal model (see later)

Query Answering under Count Semantics is Hard

Proposition ([Kostylev and Reutter, 2015])

COUNT *is* coNP*-complete for DL-Lite*$_{core}$ *and CQs.*

- Actually, for this problem we lose two desiderable properties when it comes to tractability:
  - Negative information affects the answers to a query
  - There is no universal model (see later)
- However, the CQ used in the reduction is disconnected, which is very unnatural

Query Shapes

- We study the following basic fragments of CQs:
  - Atomic Queries: AQ
  - Acyclic CQs: CQ^{A}
  - Connected CQs: CQ^{C}
  - Linear (Non-branching) CQs: CQ^{L}
  - Rooted CQs: CQ^{R}
- We also study a number of combinations of these fragments

Tractability

Proposition

COUNT is in PTIME in data complexity for DL-Lite$_{pos}^{\mathcal{H}^-\mathcal{N}^-}$ and connected, linear CQs (CQ^{CL}).

The superscript "−" over $\mathcal{H}$ limits the interaction between role subsumption and existential restrictions:

$$\text{If } B \sqsubseteq \ \geqslant_n R_1 \in \mathcal{T}, \text{ then } R_1 \sqsubseteq R_2 \notin \mathcal{T}$$

Proof (Sketch)

We start by showing it for *DL-Lite*$_{pos}^{\mathcal{H}^-}$. We consider the set matches$(q, \mathcal{I}_{can}^{\mathcal{K}})$ of all matches for q over the canonical interpretation, and consider all constant-preserving functions minimizing the size of such set. Then, due to the limited expressivity of *DL-Lite*$_{pos}^{\mathcal{H}^-\mathcal{N}^-}$, it can be proved that one of these functions is such that:

$$|f(\text{matches}(q, \mathcal{I}_{can}^{\mathcal{K}}))| = |\text{matches}(q, f(\mathcal{I}_{can}^{\mathcal{K}}))|$$

It can be shown that $|f(\text{matches}(q, \mathcal{I}_{can}^{\mathcal{K}}))|$ can be computed in polynomial time in $|\mathcal{A}|$.

For *DL-Lite*$_{pos}^{\mathcal{H}^-\mathcal{N}^-}$ the strategy is similar, however we associate to each anonymous object a cardinality (given by the number restrictions in the TBox).

Subcase 1: Linear but Disconnected (I)

Proposition

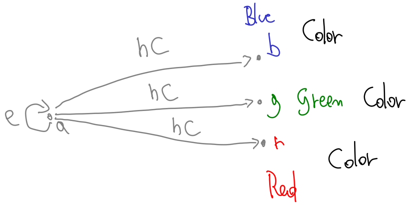
COUNT *is* CONP*-hard in data complexity for DL-Lite$_{pos}$ and acyclic, linear, but disconnected CQs* (CQ$^{\text{AL}}$)*.*

Proof (Sketch)

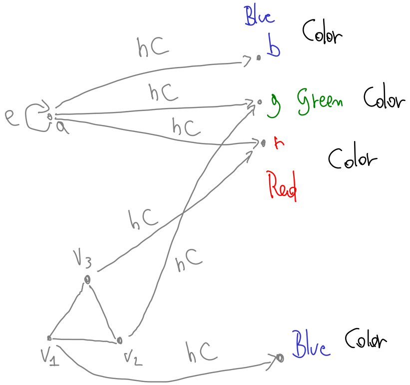
This is a direct adaptation of the proof by [Kostylev and Reutter, 2015] a reduction from the co-3-colorability problem to COUNT.

- $\mathcal{A} = \{\text{Vertex}(v) \mid v \in V\} \cup \{\text{edge}(v_1, v_2) \mid (v_1, v_2) \in E\} \cup \{\text{Blue}(\text{b}), \text{Green}(\text{g}), \text{Red}(\text{r}), \text{hasColor}(\text{a}, \text{b}), \text{hasColor}(\text{a}, \text{g}), \text{hasColor}(\text{a}, \text{r}), \text{edge}(\text{a}, \text{a})\}$
- $\mathcal{T} = \{\text{Vertex} \sqsubseteq \exists\text{hasColor}, \exists\text{hasColor}^- \sqsubseteq \text{Color}\}$
- $q() \leftarrow \text{Color}(c), \text{edge}(v_1, v_2), \text{hasColor}(v_1, c_1), \text{hasColor}(v_2, c_2), \text{Blue}(c_1), \text{Blue}(c_2), \text{edge}(v_3, v_4), \text{hasColor}(v_3, c_3), \text{hasColor}(v_4, c_4), \text{Green}(c_3), \text{Green}(c_4), \text{edge}(v_5, v_6), \text{hasColor}(v_5, c_5), \text{hasColor}(v_6, c_6), \text{Red}(c_5), \text{Red}(c_6).$
- Then it can be verified that $4 = \text{certCard}(q, \langle \mathcal{T}, \mathcal{A} \rangle)$ iff $\mathcal{G}$ is not 3-colorable.

Example of the Reduction



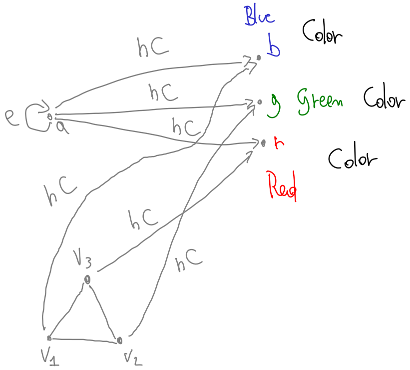
Example of the Reduction



Query

$q() \leftarrow \text{Color}(c), \text{edge}(v_1, v_2), \text{hasColor}(v_1, c_1), \text{hasColor}(v_2, c_2), \text{Blue}(c_1), \text{Blue}(c_2), \text{edge}(v_3, v_4), \text{hasColor}(v_3, c_3), \text{hasColor}(v_4, c_4), \text{Green}(c_3), \text{Green}(c_4), \text{edge}(v_5, v_6), \text{hasColor}(v_5, c_5), \text{hasColor}(v_6, c_6), \text{Red}(c_5), \text{Red}(c_6)$ 4

Example of the Reduction



Query

$q() \leftarrow$ Color(c), edge(v_1, v_2), hasColor(v_1, c_1), hasColor(v_2, c_2), Blue(c_1), Blue(c_2), edge(v_3, v_4), hasColor(v_3, c_3), hasColor(v_4, c_4), Green(c_3), Green(c_4), edge(v_5, v_6), hasColor(v_5, c_5), hasColor(v_6, c_6), Red(c_5), Red(c_6) 3

Sub-case 2: Connected but Non-linear

Proposition

COUNT *is* coNP*-hard in data complexity for DL-Lite*$^{\mathcal{H}}_{pos}$ *and acyclic, connected, but branching CQs* (CQ$^{\text{AC}}$)*.*

Proof (Sketch)

This also is a reduction from the co-3-colorability problem to COUNT. Interestingly, the number to be checked is not a fixed quantity, but is linear in the size of the graph.

Outline

Non-rewritability I

Proposition

COUNT *is* PTIME*-hard in data complexity for DL-Lite*$^{\mathcal{H}}_{core}$ *and atomic queries* (AQ)*.*

Proof (Sketch.)

Through a LOGSPACE reduction from the boolean circuit value (CVP) co-problem where all gates are NAND gates and each gate has fan-out of at-most 2.

Non-rewritability II

Proposition

COUNT *is* PTIME*-hard in data complexity for DL-Lite$^{\mathcal{H}}_{core}$ and rooted, connected, linear queries (*$\mathrm{CQ}^{\mathrm{CLR}}$*).*

Proof (Sketch)

This proof is an adaptation of the previous one.

Towards Rewritability: Universal Model

Definition (Universal Model [Nikolaou et al., 2019])

A model $\mathcal{I}$ of a KB $\mathcal{K}$ is *universal* for a class of queries $\mathcal{Q}$ iff $\text{ans}(q, \mathcal{I}) = \mathit{cert}(q, \mathcal{K})$ holds for every $q \in \mathcal{Q}$.

Do Universal Models Exist? (I)

Alert!

Under count semantics, the universal model is lost even for very basic DL-Lite members and very restrictive fragments of CQs.

Example

DL-Lite$_{pos}$ does not admit a universal model w.r.t. atomic queries, already.

$$\mathcal{A} = \left\{ \begin{array}{l} A(a), B(b), \\ P(a, b) \end{array} \right\} \qquad \mathcal{T} = \left\{ \begin{array}{l} A \sqsubseteq \exists Q, \\ \exists Q^{-} \sqsubseteq B \end{array} \right\}$$

Do Universal Models Exist? (I)

Alert!

Under count semantics, the universal model is lost even for very basic DL-Lite members and very restrictive fragments of CQs.

Example

DL-Lite$_{pos}$ does not admit a universal model w.r.t. atomic queries, already.

$$\mathcal{A} = \left\{ \begin{array}{l} A(a), B(b), \\ P(a, b) \end{array} \right\} \qquad \mathcal{T} = \left\{ \begin{array}{l} A \sqsubseteq \exists Q, \\ \exists Q^- \sqsubseteq B \end{array} \right\}$$

Queries:

- $q() \leftarrow B(y)$

Do Universal Models Exist? (I)

Alert!

Under count semantics, the universal model is lost even for very basic DL-Lite members and very restrictive fragments of CQs.

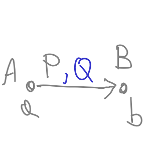
Example

DL-Lite$_{pos}$ does not admit a universal model w.r.t. atomic queries, already.

$$\mathcal{A} = \left\{ \begin{array}{l} A(a), B(b), \\ P(a,b) \end{array} \right\} \qquad \mathcal{T} = \left\{ \begin{array}{l} A \sqsubseteq \exists Q, \\ \exists Q^- \sqsubseteq B \end{array} \right\}$$

Queries:

- $q() \leftarrow B(y)$



Do Universal Models Exist? (I)

Alert!

Under count semantics, the universal model is lost even for very basic DL-Lite members and very restrictive fragments of CQs.

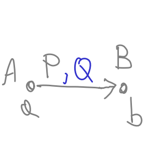
Example

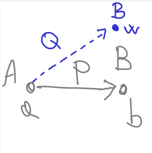
DL-Lite$_{pos}$ does not admit a universal model w.r.t. atomic queries, already.

$$\mathcal{A} = \left\{ \begin{array}{l} A(a), B(b), \\ P(a,b) \end{array} \right\} \qquad \mathcal{T} = \left\{ \begin{array}{l} A \sqsubseteq \exists Q, \\ \exists Q^{-} \sqsubseteq B \end{array} \right\}$$

Queries:

- $q() \leftarrow B(y)$
- $q() \leftarrow Q(a,b)$





Do Universal Models Exist? (II)

Proposition

DL-Lite$_{core}^{\mathcal{N}^-}$ has a universal model w.r.t. COUNT *over* CQ^{CR} *queries.*

Proof

By showing that the restricted chase [Calvanese et al., 2013], [Botoeva et al., 2010] is universal.

Rewriting Algorithm

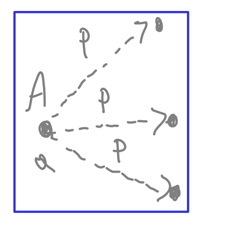
- The existence of a universal model is a hint that a rewriting algorithm might exist
- In our work we devise such a rewriting algorithm, however:
  - It is highly non-trivial (and definitely too verbose to be formally presented here)
  - It is mostly of theoretical interest, and not very practical
- The query language for the rewriting is in LOGSPACE (data complexity), and it has aggregation variables, nested aggregation, and a limited form of arithmetics
- Such language has a direct translation into SQL

Why is the Algorithm Non-trivial? Anonymous Contribution

Example

KB: $\mathcal{T} = \{A \sqsubseteq\ \geqslant_3 P\}$, $\mathcal{A} = \{A(a)\}$
Input Query: $q(x) \leftarrow P(x, y)$

The original query, part of the rewriting, looks in the ABox for all P-paths of length 1:

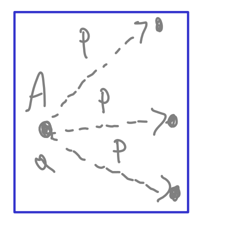


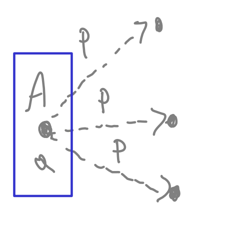
Why is the Algorithm Non-trivial? Anonymous Contribution

Example

KB: $\mathcal{T} = \{A \sqsubseteq\ \geqslant_3 P\}$, $\mathcal{A} = \{A(a)\}$
Input Query: $q(x) \leftarrow P(x, y)$

The original query, part of the rewriting, looks in the ABox for all P-paths of length 1:





Rewritten CQ $q'(x) \leftarrow A(x)$
There is a single match $\mu = \{x \mapsto a\}$ for q' over $\mathcal{A}$, which can be extended into exactly three matches for q in $ch_\infty(\mathcal{K})$, by mapping variable y into some anonymous object.

Rewriting Rationale

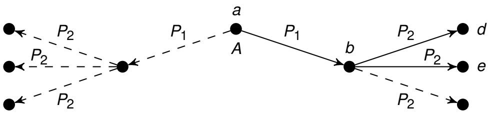
- We need to partition the queries, taking into account their anonymous contribution (i.e., number of ways a mapping can be extended into the anonymous part)
- We need to guarantee that the partitions are disjoint
- Each partition is a generalized union handling the removal of duplicate answers introduced by the rewriting itself
- The anonymous contribution needs to be computed by saturating the subsumptions in the TBox, and through an atomic decomposition of concepts and roles

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$

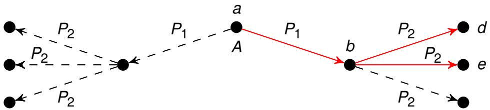
Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lll} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Batman Example

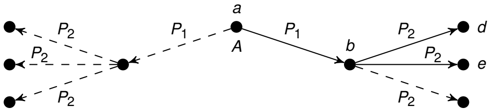
$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$ 2

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lll} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



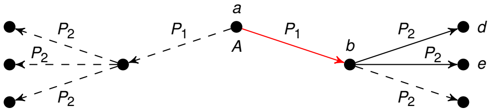
Query: $q(x) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)$ 2

GE_α:

$\langle$ $\}\rangle$

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



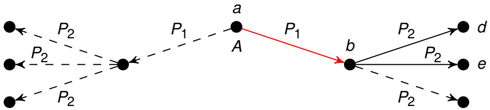
Query: $q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$ 2

GE_α:

$\langle \quad \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \quad \}\rangle$

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lll} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



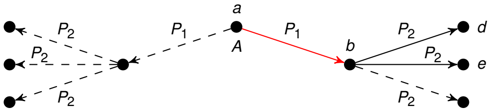
Query: $q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$ 2

GE_α:

$\langle$ $\{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=2} P_2(y_1, z)\}\rangle$

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



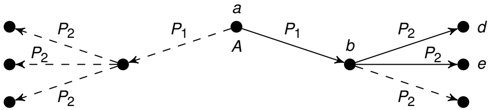
Query: $q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$ 2

GE_α:

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=2} P_2(y_1, z)\}\rangle$ 1

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)$ 2

GE_α:

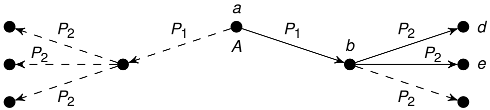
$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 0), \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=0} P_2(y_1,z)\}\rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 1), \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=1} P_2(y_1,z)\}\rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=2} P_2(y_1,z)\}\rangle$ 1

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$ 2

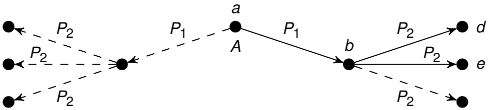
$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 0), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=0} P_2(y_1, z)\}\rangle$

GE_α: $\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 1), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=1} P_2(y_1, z)\}\rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=2} P_2(y_1, z)\}\rangle$ 1

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$ 2

GE_α:

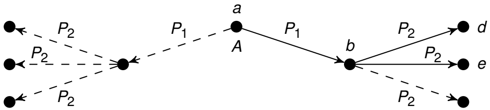
$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 0), \left\{ \begin{array}{l} q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=0} P_2(y_1, z) \\ q(x : y_1) \leftarrow A(x), P_1(x, y_1) \exists_z^{=0} P_2(y_1, z) \end{array} \right\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 1), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=1} P_2(y_1, z)\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=2} P_2(y_1, z)\} \rangle$ 1

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lcl} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)$ 2

GE_α:

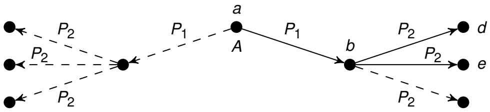
$$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 0), \left\{ \begin{array}{l} q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=0} P_2(y_1,z) \\ q(x : y_1) \leftarrow A(x), P_1(x,y_1) \exists_z^{=0} P_2(y_1,z) \end{array} \right\} \rangle$$

$$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 1), \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=1} P_2(y_1,z)\} \rangle$$

$$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=2} P_2(y_1,z)\} \rangle \quad 1$$

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lll} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)$ 2

GE_α:

$$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 0), \left\{ \begin{array}{l} q(x:y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=0} P_2(y_1,z) \\ q(x:y_1) \leftarrow A(x), P_1(x,y_1) \exists_z^{=0} P_2(y_1,z) \end{array} \right\} \rangle$$

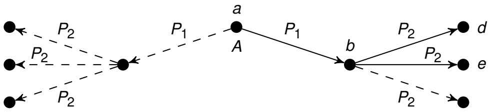
$$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 1), \{ q(x:y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=1} P_2(y_1,z) \} \rangle$$

$$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{ q(x:y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=2} P_2(y_1,z) \} \rangle$$ 1

GE_β:

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lll} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x,y_1), P_2(y_1,y_2)$ 2

GE_α:

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 0), \left\{ \begin{array}{l} q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=0} P_2(y_1,z) \\ q(x : y_1) \leftarrow A(x), P_1(x,y_1) \exists_z^{=0} P_2(y_1,z) \end{array} \right\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 1), \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=1} P_2(y_1,z)\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{q(x : y_1) \leftarrow A(x), P_1(x,y_1), P_1(\_,y_1), \exists_z^{=2} P_2(y_1,z)\} \rangle$ 1

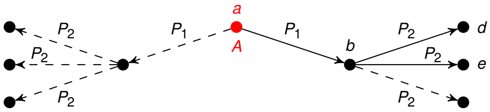
GE_β:

$\langle Q(x, \mathrm{cnt}(y_1) \cdot (2 - 0) \cdot 3), \{q(x) \leftarrow A(x), \exists_y^{=0} P_1(x,y)\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot (2 - 1) \cdot 3), \{q(x) \leftarrow A(x), \exists_y^{=1} P_1(x,y)\} \rangle$

Batman Example

$$\mathcal{T} = \left\{ \begin{array}{lll} A & \sqsubseteq & \geqslant_2 P_1, \\ \exists P_1^- & \sqsubseteq & \geqslant_3 P_2 \end{array} \right\}, \quad \mathcal{A} = \left\{ \begin{array}{l} A(a),\ P_1(a,b), \\ P_2(b,d),\ P_2(b,e) \end{array} \right\}$$



Query: $q(x) \leftarrow A(x), P_1(x, y_1), P_2(y_1, y_2)$ 2

GE_α:

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 0), \left\{ \begin{array}{l} q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=0} P_2(y_1, z) \\ q(x : y_1) \leftarrow A(x), P_1(x, y_1) \exists_z^{=0} P_2(y_1, z) \end{array} \right\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 1), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=1} P_2(y_1, z)\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot 3 - 2), \{q(x : y_1) \leftarrow A(x), P_1(x, y_1), P_1(\_, y_1), \exists_z^{=2} P_2(y_1, z)\} \rangle$ 1

GE_β:

$\langle Q(x, \mathrm{cnt}(y_1) \cdot (2 - 0) \cdot 3), \{q(x) \leftarrow A(x), \exists_y^{=0} P_1(x, y)\} \rangle$

$\langle Q(x, \mathrm{cnt}(y_1) \cdot (2 - 1) \cdot 3), \{q(x) \leftarrow A(x), \exists_y^{=1} P_1(x, y)\} \rangle$ 3

Outline

- The Setting
- Tractability and Intractability
- Rewritability and Non-rewritability
- Conclusions and Future Directions

Recap of Results

	AQ,CQ^{CL}	CQ^{AC}	CQ^{CLR},CQ^{CR}	CQ^{AL}	CQ
$\mathit{DL\text{-}Lite}_{pos}$	P	coNP	L	coNP-c	coNP-c
$\mathit{DL\text{-}Lite}_{pos}^{\mathcal{H}}$	PTime	coNP-c	coNP	coNP-c	coNP-c
$\mathit{DL\text{-}Lite}_{pos}^{\mathcal{H}^-\mathcal{N}^-}$	PTime	coNP-c	coNP	coNP-c	coNP-c
$\mathit{DL\text{-}Lite}_{core}$	coNP	coNP	L	coNP-c	coNP-c
$\mathit{DL\text{-}Lite}_{core}^{\mathcal{N}^-}$	coNP	coNP	L	coNP-c	coNP-c
$\mathit{DL\text{-}Lite}_{core}^{\mathcal{H}}$	PTime-h/coNP	PTime-h/coNP	PTime-h/coNP	coNP-c	coNP-c

Table: Summary of complexity results ('-h' stands for '-hard', and '-c' for '-complete'). New bounds proved here are in blue, bounds that directly follow in green, and already known bounds in black.

- Open questions:
  - Is the *P*-membership result for $\mathit{DL\text{-}Lite}_{core}^{\mathcal{H}^-\mathcal{N}^-}$ and AQ,CQ^{CL} tight?
  - Does rewritability hold on $\mathit{DL\text{-}Lite}_{core}^{\mathcal{H}^-\mathcal{N}^-}$ and rooted queries?
  - What if we consider numbers in the TBox for data-complexity?
    - Our rewriting produces a query whose size is exponential in such numbers, when these are encoded in binary

Thanks.

Thanks.

References I

[Bienvenu and Ortiz, 2015] Bienvenu, M. and Ortiz, M. (2015).
Ontology-mediated query answering with data-tractable description logics.
In *Reasoning Web. Web Logic Rules – 11th Int. Summer School Tutorial Lectures (RW)*, volume 9203 of *Lecture Notes in Computer Science*, pages 218–307. Springer.

[Botoeva et al., 2010] Botoeva, E., Artale, A., and Calvanese, D. (2010).
Query rewriting in $\textit{DL-Lite}^{\mathcal{HN}}_{\textit{horn}}$.
In *Proc. of the 23rd Int. Workshop on Description Logics (DL)*, volume 573 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, pages 267–278.

[Calvanese et al., 2013] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2013).
Data complexity of query answering in description logics.
Artificial Intelligence, 195:335–360.

[Kostylev and Reutter, 2015] Kostylev, E. V. and Reutter, J. L. (2015).
Complexity of answering counting aggregate queries over DL-Lite.
J. of Web Semantics, 33:94–111.

References II

[Nikolaou et al., 2019] Nikolaou, C., Kostylev, E. V., Konstantinidis, G., Kaminski, M., Cuenca Grau, B., and Horrocks, I. (2019).
Foundations of ontology-based data access under bag semantics.
Artificial Intelligence, 274:91–132.

[Poggi et al., 2008] Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Rosati, R. (2008).
Linking data to ontologies.
J. on Data Semantics, X:133–173.