On the Succinctness of Query Rewriting for Datalog

Shqiponja Ahmetaj\textsuperscript{1} \quad Andreas Pieris\textsuperscript{2}

\textsuperscript{1}Institute of Logic and Computation, TU Wien, Austria

\textsuperscript{2}School of Informatics, University of Edinburgh, UK

Foundational Challenges in Data and Knowledge Management, Vienna, March 23, 2018
Ontology-Based Query Answering

Certain-Answers\((q, D, \Sigma)\) = \{ \(c_1, \ldots, c_n\) \in \text{dom}(D)^n \mid D \land \Sigma \models q(c_1, \ldots, c_n) \}
Ontology-Mediated Queries

$$Q = (\Sigma, q(x_1, \ldots, x_n))$$

$$Q(D) = \text{Certain-Answers}(q, D, \Sigma)$$
Scalable OMQ Evaluation

Database

ontology-mediated query (OMQ)

\[ Q = (\Sigma, q(x_1, \ldots, x_n)) \]

Exploit standard RDBMSs - efficient technology for answering queries
Query Rewriting

\[ Q = (\Sigma, q(x_1, \ldots, x_n)) \]

\[ Q_{\text{rew}}(x_1, \ldots, x_n) \]

a query that can be executed by exploiting existing database technology

for every database \( D \) : \( Q(D) = Q_{\text{rew}}(D) \)
Query Rewriting: An Example

\[ Q = (\Sigma, q) \]

\[ Q_{rew} = \exists x (\text{Person}(x) \land \text{HasFather}(\text{John}, x)) \lor \text{Person}(\text{John}) \]
Query Rewritability

\[(L, CQ)\]

- an ontology language
- the class of conjunctive queries
  (fragment of first-order logic)

**Definition:** An OMQ language \( O \) is **QL-Rewritable** if every \( Q \in O \) is QL-Rewritable

First-order (FO), \( \exists FO^+ \), Non-recursive Datalog (NDL), UCQ or Datalog
Query Rewritability: The Main Questions

1. Can we isolate meaningful OMQ languages that are QL-Rewritable?

2. What is the price of QL-rewriting?

...have been extensively studied for DL- and rule-based OMQ languages
Tuple-Generating Dependencies (TGDs)

(a.k.a. existential rules or Datalog\(\pm\) rules)

\[ \forall x \forall y (\varphi(x,y) \rightarrow \exists z \varphi(x,z)) \]

\((\text{TGD}, \text{CQ})\)
The Guarded Family

**Weakly-Guarded**

one body-atom contains all the harmful $\forall$-variables

$$R(w), \varphi(x,y) \rightarrow \exists z \psi(x,z) \quad - \quad w \subseteq \{x,y\}$$


**Guarded**

one body-atom contains all the $\forall$-variables

$$R(x,y), \varphi(x,y) \rightarrow \exists z \psi(x,z)$$


**Linear**

one body-atom

$$R(x,y) \rightarrow \exists z \psi(x,z)$$

The Guarded Family

**Theorem:** It holds that

1. (Linear, CQ) is UCQ-Rewritable
2. (Guarded, CQ) is not FO-Rewritable, but is Datalog-Rewritable
3. (Weakly-Guarded, CQ) is not Datalog-Rewritable
(Guarded, CQ) is not FO-Rewritable

\[ Q = \{\{R(x,y), P(y) \rightarrow P(x)\}, \ P(c_n)\} \]

\[ D \supseteq \{P(c_1)\}, \text{ and contains no other } P\text{-atom} \]

\[ Q_{rew} \text{ has to check for the existence of an } R\text{-path in } D \text{ of unbounded length} \]

\[ c_n \xrightarrow{R} \#_{n-1} \xrightarrow{R} \#_{n-2} \xrightarrow{R} \ldots \xrightarrow{R} \#_2 \xrightarrow{R} c_1 \]

compute the transitive closure of R - not possible via a first-order query
(Weakly-Guarded, CQ) is not Datalog-Rewritable

Evaluation of (Weakly-Guarded, CQ) queries is

EXPTIME-complete in data complexity

...in fact, (Weakly-Guarded\textsuperscript{\textneg stratified}, CQ) = EXPTIME, even w/o an order

[Gottlob, Rudolph & Šimkus, PODS 2014]
The Guarded Family

**Theorem**: It holds that

1. \((\text{Linear}, \text{CQ})\) is UCQ-Rewritable
2. \((\text{Guarded}, \text{CQ})\) is not FO-Rewritable, but is Datalog-Rewritable
3. \((\text{Weakly-Guarded}, \text{CQ})\) is not Datalog-Rewritable
The Guarded Family

**Theorem:** It holds that

1. *(Linear,CQ)* is UCQ-Rewritable

2. *(Guarded,CQ)* is not FO-Rewritable, but is Datalog-Rewritable

3. *(Weakly-Guarded,CQ)* is not Datalog-Rewritable
(Linear, CQ) is UCQ-Rewritable

Via a resolution-based algorithm - XRewrite

ALGORITHM 1: The algorithm XRewrite

Input: a CQ $q$ over a schema $\mathcal{R}$ and a set $\Sigma$ of TGDs over $\mathcal{R}$
Output: the perfect rewriting of $q$ w.r.t. $\Sigma$

\begin{verbatim}
   \begin{algorithm}
   \begin{tabular}{l}
   \textbf{Input:} a CQ $q$ over a schema $\mathcal{R}$ and a set $\Sigma$ of TGDs over $\mathcal{R}$
   \\
   \textbf{Output:} the perfect rewriting of $q$ w.r.t. $\Sigma$
   \\
   \textbf{Algorithm XRewrite:}
   \\
   $i := 0$;
   \\
   $Q_{\text{rew}} := \{(q, r, u)\}$;
   \\
   \textbf{repeat}
   \\
   $Q_{\text{rew}} := Q_{\text{rew}}$;
   \\
   \textbf{foreach} $(q, x, u) \in Q_{\text{rew}}$, where $x \in \{r, f\}$ \textbf{do}
   \\
   \textbf{foreach} $\sigma \in \Sigma$ \textbf{do}
   \\
   \textbf{/* rewriting step */}
   \\
   \textbf{foreach} $S \subseteq \text{body}(q)$ such that $\sigma$ is applicable to $S$ \textbf{do}
   \\
   \begin{tabular}{l}
   \textbf{if} there is no $(q^*, r, \ast) \in Q_{\text{rew}}$ such that $q^* \simeq q^*$ \textbf{then}
   \\
   $Q_{\text{rew}} := Q_{\text{rew}} \cup \{(q', r, u)\}$;
   \\
   \end{tabular}
   \\
   $i := i + 1$;
   \\
   $q' := \gamma_{S, \ast'}(q[S/body(\sigma')])$;
   \\
   \textbf{end}
   \\
   \textbf{end}
   \\
   \textbf{end}
   \\
   \\
   \textbf{end}
   \\
   \textbf{/* factorization step */}
   \\
   \textbf{foreach} $S \subseteq \text{body}(q)$ which is factorizable w.r.t. $\sigma$ \textbf{do}
   \\
   $q' := \gamma_S(q)$;
   \\
   \textbf{if} there is no $(q^*, r, \ast) \in Q_{\text{rew}}$ such that $q^* \simeq q^*$ \textbf{then}
   \\
   $Q_{\text{rew}} := Q_{\text{rew}} \cup \{(q', r, u)\}$;
   \\
   \textbf{end}
   \\
   \textbf{end}
   \\
   \textbf{end}
   \\
   \textbf{/* query $q$ is now explored */}
   \\
   $Q_{\text{rew}} := (Q_{\text{rew}} \setminus \{(q, x, u)\}) \cup \{(q, x, e)\}$;
   \\
   \textbf{end}
   \\
   \textbf{until} $Q_{\text{rew}} = Q_{\text{rew}}$;
   \\
   $Q_{\text{rew}} := \emptyset$;
   \\
   return $Q_{\text{rew}}$;
   \end{algorithm}
\end{verbatim}

applicability condition for TGDs

apply useful reduction steps, but only useful ones

(Linear, CQ) is UCQ-Rewritable

Via a resolution-based algorithm - XRewrite

Given an OMQ $Q = (\Sigma, q)$ from (Linear, CQ)

1. The height of $XRewrite(Q)$ is at most $|q|

2. The size of $XRewrite(Q)$ is at most $\#pred(\Sigma)^{|q|} \cdot (\text{arity}(\Sigma) \cdot |q|)^{\text{arity}(\Sigma) \cdot |q|}$

worst-case optimal

Lower Bound for \( \text{(Linear, CQ)} \)

\[
\Sigma = \{ R_i(x) \rightarrow P_i(x) \}_{i \in \{1, \ldots, n\}} \quad \text{q} = \exists x (P_1(x) \land \ldots \land P_n(x))
\]

\[
\exists x \ P_1(x) \land \ldots \land P_n(x)
\]

\[
P_1(X) \lor R_1(X) \quad P_n(X) \lor R_n(X)
\]

\( \Rightarrow \) we need to consider \( 2^n \) disjuncts
Theorem: For (Linear, CQ) there is

- No $\exists$FO$^+/\text{NDL}$-rewriting of polynomial size
- No FO-rewriting of polynomial size (unless the PH collapses)

Proof: Via succinctness of monotone Boolean circuits

NOTE: The above proof exploits databases with a single domain element
Two Domain Elements

\[ Q = (\Sigma, q(x_1, \ldots, x_n)) \]

rewrite in polynomial time

\[ Q_{\text{rew}}(x_1, \ldots, x_n) \]

for every database \( D_{01} : Q(D_{01}) = Q_{\text{rew}}(D_{01}) \)

\[ \subseteq \{ \text{Zero}(0), \text{One}(1) \} \]
Polynomial Rewritings

... assuming two domain elements

**Theorem:** A \((\text{Linear, CQ})\) query can be rewritten in polynomial time as:

- An \(\exists FO^+/\text{NDL}\) query for bounded arity predicates
- An \(\text{FO}\) query for arbitrary signatures

**Proof:**

- Bounded arity signatures - via the polynomial witness property
- Arbitrary signatures - via proof generators
Polynomial Witness Property (PWP)

**Definition:** $(L, CQ)$ enjoys the PWP if: there exists a polynomial $\text{pol}(\cdot)$ such that for every $Q = (\Sigma, q(x)) \in (L, CQ)$, database $D$, and $t \in \text{dom}(D)^{|x|}$

$$t \in Q(D) \Rightarrow q(t) \text{ can be entailed after } \text{pol}(|\Sigma|, |q|) \text{ chase steps}$$

**Theorem:** PWP $\Rightarrow \exists \text{FO}^+/\text{NDL}$-rewritings constructible in polynomial time, focusing on databases with at least two constants

Witnesses and Linearity

\[ 0 = \{ \} \quad q = \exists z \exists o \text{ Number}(o, \ldots, o, z, o) \]

\[
\{ \text{Number}(x_1, \ldots, x_{n-i}, z, o, \ldots, o, z, o) \rightarrow \text{Number}(x_1, \ldots, x_{n-i}, o, z, \ldots, z, z, o) \}_{i \in \{1, \ldots, n\}}
\]

\[
\begin{align*}
0 & \quad \text{Number}(0, \ldots, 0, 0, 0, 1) \\
1 & \quad \text{Number}(0, \ldots, 0, 1, 0, 1) \\
2 & \quad \text{Number}(0, \ldots, 1, 0, 0, 1) \\
2^n & \quad \text{Number}(1, \ldots, 1, 1, 0, 1)
\end{align*}
\]
Proof Generator

\[ q = \exists x \exists y \exists z \exists w \ (P(x, a, y) \land P(z, y, b) \land P(w, c, b)) \]

\[ \alpha = (\ldots z_1 \ldots) \]
\[ \beta = (\ldots z_2 \ldots) \]
\[ \gamma = (\ldots z_4 \ldots) \]
\[ \delta = (\ldots z_3 \ldots) \]

\[ \text{chase forest} \]

\[ \beta \]
\[ \delta \]

\[ h, \ \{\alpha, \beta, \gamma, \delta\}, \]

\[ D \]

[Gotlob, Manna & P., IJCAI 2015]
Proof Generator

\[ k = (|q| + 1) \cdot (2 \cdot \text{arity})^\text{arity} \]

\[ q = \exists x \exists y \exists z \exists w \ (P(x,a,y) \land P(z,y,b) \land P(w,c,b)) \]

check via an FO query whether a proof generator exists

[Gotlob, Manna & P., IJCAI 2015]
Polynomial Rewritings

... assuming two domain elements

**Theorem:** A (Linear, CQ) query can be rewritten in polynomial time as:

- An ∃FO⁺/NDL query for bounded arity predicates
- An FO query for arbitrary signatures

**Proof:**

- Bounded arity signatures - via the polynomial witness property
- Arbitrary signatures - via proof generators
The Guarded Family

**Theorem:** It holds that

1. *(Linear,CQ)* is UCQ-Rewritable

2. *(Guarded,CQ)* is not FO-Rewritable, but is Datalog-Rewritable

3. *(Weakly-Guarded,CQ)* is not Datalog-Rewritable
(Guarded, CQ) is Datalog-Rewritable

Via inference rules - inspired by DLs

\[
\begin{align*}
\alpha \to \beta \land A & \quad \frac{A \text{ has no existential variables}}{\alpha \to A} \\
\alpha \to \beta & \quad \frac{\gamma_1 \land \gamma_2 \to \delta}{\alpha \land h(\gamma_1) \to \beta \land h(\delta)} \\
& \quad \frac{\gamma_1 \land \gamma_2 \to \delta \text{ is a Datalog rule,} \quad h \text{ is a homomorphism from}}{
\quad \quad \gamma_2 \text{ to } \beta \text{ with } \text{vars}(h(\gamma_1)) \subseteq \text{vars}(\alpha).}
\end{align*}
\]
(Guarded, CQ) is Datalog-Rewritable

Via inference rules - inspired by DLs

Given an OMQ $Q = (\Sigma, q)$ from (Guarded, CQ),

the size of the Datalog rewriting is at most $2^{\left(\#\text{pred}(\Sigma) \cdot \#\text{body-vars}(\Sigma)^{\text{arity}(\Sigma)}\right)}$

worst-case optimal?

[Gottlob, Rudolph & Šimkus, PODS 2014]
Polynomial Rewritings

... assuming two domain elements

**Theorem:** A (Guarded, Full CQ) or (Guarded, Acyclic CQ) query over bounded arity predicates can be rewritten in polynomial time as a Datalog query

**Proof:** Via types

- Build all possible types
- Mark “bad” types
- From marked types to Datalog rules that capture ground consequences
Wrap Up

\[(\text{Linear,CQ}) \quad \text{EXPTIME} \quad \text{UCQ}\]

\[(\text{Linear,CQ}) \quad \text{PTIME} \quad \exists \text{FO}^+, \text{NDL, FO}\]

\[(\text{Linear,CQ}) \quad \text{PTIME} \quad \exists \text{FO}^+, \text{NDL} \quad \text{bounded arity, \{0,1\}}\]

\[(\text{Linear,CQ}) \quad \text{PTIME} \quad \text{FO} \quad \{0,1\}\]
Wrap Up

\[(\text{Guarded, CQ}) \xrightarrow{\text{2EXPTIME}} \text{FO}\]

\[(\text{Guarded, CQ}) \xrightarrow{\text{EXPTIME}} \text{Datalog}\]

\[(\text{Guarded, CQ}) \xrightarrow{\text{bounded arity}} \text{Datalog}\]

\[(\text{Guarded, FCQ/ACQ}) \xrightarrow{\text{PTIME}} \text{Datalog}\]
Open Problems

- (Linear, CQ) $\xrightarrow{\text{PTIME}}$ UCQ
  - bounded arity, \{0,1\}

- (Linear, CQ) $\xrightarrow{\text{PTIME}}$ NDL
  - \{0,1\}

- (Guarded, CQ) $\xrightarrow{\text{PTIME}}$ Datalog
  - bounded arity, \{0,1\}

Thank You!