Worst-Case Optimal Querying of Very Expressive Description Logics with Path Expressions and Succinct Counting*

Bartosz Bednarczyk[1,2] and **Sebastian Rudolph**[2]
[1]Institute of Computer Science, University of Wrocław
[2]Computational Logic Group, TU Dresden
bartosz.bednarczyk@cs.uni.wroc.pl, sebastian.rudolph@tu-dresden

Abstract

Among the most expressive knowledge representation formalisms are the description logics of the $\mathcal{Z}$ family. For well-behaved fragments of $\mathcal{ZOIQ}$, entailment of positive two-way regular path queries is well known to be 2ExpTime-complete under the proviso of unary encoding of numbers in cardinality constraints. We show that this assumption can be dropped without an increase in complexity and ExpTime-completeness can be achieved when bounding the number of query atoms, using a novel reduction from query entailment to knowledge base satisfiability. These findings allow to strengthen other results regarding query entailment and query containment problems in very expressive description logics. Our results also carry over to $\mathcal{GC}^2$, the two-variable guarded fragment of first-order logic with counting quantifiers, for which hitherto only conjunctive query entailment has been investigated.

1 Introduction

Recent years have seen a convergence of the fields of logic-based knowledge representation (KR) and databases, with querying over knowledge bases (KBs, aka ontologies) as the archetypical inferencing task considered. Thereby, the query languages of interest have evolved from plain conjunctive queries to more expressive formalisms with path-navigational components such as the *positive two-way regular path queries* (P2RPQs) considered here. One popular and practically used way of specifying the to-be-queried knowledge bases is via *description logics* (DLs), whose most expressive exemplars feature advanced constructors for roles and path expressions. Certainly, the DLs from the $\mathcal{Z}$ family [Calvanese *et al.*, 2009] are among the most powerful KR formalisms on the verge of decidability. For its most expressive proponent, $\mathcal{ZOIQ}$, featuring nominals ($\mathcal{O}$), role inverses ($\mathcal{I}$), and number restrictions ($\mathcal{Q}$), querying is undecidable [Ortiz *et al.*, 2010; Rudolph, 2016] and even decidability of KB satisfiability is open, owing to the intricate interplay of the three mentioned features, but restricting the interaction of $\mathcal{O}$, $\mathcal{I}$, and $\mathcal{Q}$ (or excluding one of the features altogether) leads to beneficial model-theoretic properties, which give rise to upper bounds of ExpTime for KB satisfiability and 2ExpTime for querying, established using elaborate automata techniques [Calvanese *et al.*, 2009]. While the first bound holds under the assumption of binary encoding of number restrictions, the proof for the second one requires unary encoding.

In our paper, we overcome this restriction. Leveraging said model-theoretic property of the considered DLs, we provide a novel reduction from the query entailment problem $\mathcal{K} \models q$ to the problem of checking unsatisfiability of some other knowledge base $\mathcal{K}_{\neg q}$. Thereby, the size of the latter is exponential in the total size of q and $\mathcal{K}$ (even when assuming binary encoding), leading to a 2ExpTime-complete complexity. Our technique also yields the new insight that the complexity drops to ExpTime, when the number of atoms in q is bounded by a constant. The obtained results allow to correspondingly improve a number of previous results on query containment and can be transferred to DLs from the $\mathcal{SR}$ family.

Reaching out to the community researching decidable fragments of first-order logic, we show that our results also extend to entailment of P2RPQs by $\mathcal{GC}^2$ databases, i.e. sets of unary and binary ground facts in the presence of a sentence of the guarded two-variable fragment with counting as defined by Pratt-Hartmann (2007).

2 Preliminaries

Description Logics

We will briefly recap syntax and semantics of the very expressive DL $\mathcal{ZOIQ}$ and its relevant sublogics following Calvanese *et al.* (2009). We assume a fixed signature consisting of sets $\mathbf{N_I}$ of *individual names*, $\mathbf{N_C}$ of *concept names*, and $\mathbf{N_R}$ of *role names*. We let $\mathbf{N_C}$ contain the special concept names $\top$ and $\bot$, and $\mathbf{N_R}$ the special role names $\overline{\top}$ and $\overline{\bot}$. The following EBNF grammar defines *atomic concepts* B, *concepts* C, *atomic roles* R, *simple roles* S and *roles* T, with $o, a, b \in \mathbf{N_I}$, $A \in \mathbf{N_C}$, $P \in \mathbf{N_R} \setminus \{\overline{\top}\}$:

$$\begin{array}{lll} B & ::= & A \mid \{o\} \\ C & ::= & B \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall T.C \mid \exists T.C \mid \\ & & \geqslant nS.C \mid \leqslant nS.C \mid \exists S.\mathsf{Self} \\ R & ::= & P \mid P^- \\ S & ::= & R \mid S \cap S \mid S \cup S \mid S \setminus S \\ T & ::= & \overline{\top} \mid S \mid T \cup T \mid T \circ T \mid T^* \mid id(C) \end{array}$$

*Work supported by "Diamentowy Grant" no. DI2017 006447 and by ERC Consolidator Grant 771779 (DeciGUT).

Name	Syntax	Semantics
top	$\top$	$\Delta^\mathcal{I}$
bottom	$\bot$	$\emptyset$
nominal	$\{o\}$	$\{o^\mathcal{I}\}$
negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
conc. intersection	$C_1 \sqcap C_2$	$C_1^\mathcal{I} \cap C_2^\mathcal{I}$
conc. union	$C_1 \sqcup C_2$	$C_1^\mathcal{I} \cup C_2^\mathcal{I}$
univ. restriction	$\forall T.C$	$\{x \mid \forall y.(x,y) \in T^\mathcal{I} \to y \in C^\mathcal{I}\}$
exist. restriction	$\exists T.C$	$\{x \mid \exists y.(x,y) \in T^\mathcal{I} \wedge y \in C^\mathcal{I}\}$
qualified number	$\leqslant nS.C$	$\{x \mid \#\{y \in C^\mathcal{I} \mid (x,y) \in S^\mathcal{I}\} \leq n\}$
restriction	$\geqslant nS.C$	$\{x \mid \#\{y \in C^\mathcal{I} \mid (x,y) \in S^\mathcal{I}\} \geq n\}$
Self concept	$\exists S.\mathsf{Self}$	$\{x \mid (x,x) \in S^\mathcal{I}\}$
universal role	$\overline{\top}$	$\Delta^\mathcal{I} \times \Delta^\mathcal{I}$
bottom role	$\overline{\bot}$	$\emptyset$
inverse role	P^-	$\{(x,y) \mid (y,x) \in P^\mathcal{I}\}$
role intersection	$S_1 \cap S_2$	$S_1^\mathcal{I} \cap S_2^\mathcal{I}$
role union	$S_1 \cup S_2$	$S_1^\mathcal{I} \cup S_2^\mathcal{I}$
role difference	$S_1 \setminus S_2$	$S_1^\mathcal{I} \setminus S_2^\mathcal{I}$
role concatenation	$T_1 \circ T_2$	$T_1^\mathcal{I} \circ T_2^\mathcal{I}$
Kleene star	T^*	$\bigcup_{i \geq 0} (T^\mathcal{I})^i$
concept test	$id(C)$	$\{(x,x) \mid x \in C^\mathcal{I}\}$

Table 1: Semantics of concepts and (simple) roles in $\mathcal{ZOIQ}$.

Axiom α	$\mathcal{I} \models \alpha$, if	
$C \sqsubseteq D$	$C^\mathcal{I} \subseteq D^\mathcal{I}$	TBox $\mathcal{T}$
$C(a)$	$a^\mathcal{I} \in C^\mathcal{I}$	ABox $\mathcal{A}$
$R(a,b)$	$(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$	
$a \approx b$	$a^\mathcal{I} = b^\mathcal{I}$	
$a \not\approx b$	$a^\mathcal{I} \neq b^\mathcal{I}$	

Table 2: Syntax and semantics of $\mathcal{ZOIQ}$ axioms.

An *assertion* is of the form $C(a)$, $R(a,b)$, $a \approx b$, or $a \not\approx b$, a *general concept inclusion* (GCI) has the form $C_1 \sqsubseteq C_2$. We use $C_1 \equiv C_2$ as abbreviation for $C_1 \sqsubseteq C_2$, $C_2 \sqsubseteq C_1$. A $\mathcal{ZOIQ}$ *knowledge base* (short: KB) $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ consists of a finite set $\mathcal{A}$ (called *ABox*) of assertions and a finite set $\mathcal{T}$ (called *TBox*) of GCIs.

The semantics of $\mathcal{ZOIQ}$ is defined via *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ composed of a non-empty set $\Delta^\mathcal{I}$ called the *domain of* $\mathcal{I}$ and a function $\cdot^\mathcal{I}$ mapping individual names to elements of $\Delta^\mathcal{I}$, concept names to subsets of $\Delta^\mathcal{I}$, and role names to subsets of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$. This mapping is extended to concepts, simple roles, and roles (cf. Table 1) and finally used to define *satisfaction* of assertions and GCIs (see Table 2). We say that an interpretation $\mathcal{I}$ *satisfies* a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ (or $\mathcal{I}$ is a *model* of $\mathcal{K}$, written: $\mathcal{I} \models \mathcal{K}$) if it satisfies all axioms of $\mathcal{A}$ and $\mathcal{T}$. $\mathcal{K}$ is called *satisfiable* if it has a model and *unsatisfiable* otherwise. From $\mathcal{ZOIQ}$, we obtain $\mathcal{ZIQ}$ by disallowing nominal concepts $\{o\}$, $\mathcal{ZOQ}$, by disallowing role inverses $(.)^-$ and $\mathcal{ZOI}$ by disallowing number restrictions.

Queries

We use *variables* from a countably infinite set $\mathbf{V}$. A Boolean *positive two-way regular path query* (P2RPQ) is an expression $q = \exists \vec{x}.\varphi$, where φ is a positive formula (i.e., one using $\wedge$ and $\vee$) over expressions $T(s,t)$ or $C(t)$, where T is a role, C is concept, and s and t are from $\vec{x} \cup \mathbf{N_I}$. A P2RPQ is a *conjunctive two-way regular path query* (C2RPQ) if it does not use disjunction. It is a *union of conjunctive two-way regular path queries* (UC2RPQ) if it is a disjunction of C2RPQs.[1] A *(variable) assignment* $\mathcal{Z}$ for $\mathcal{I}$ is a mapping $\mathbf{V} \to \Delta^\mathcal{I}$. For $x \in \mathbf{V}$, we set $x^{\mathcal{I},\mathcal{Z}} := \mathcal{Z}(x)$; for $c \in \mathbf{N_I}$, we set $c^{\mathcal{I},\mathcal{Z}} := c^\mathcal{I}$. $T(s,t)$ evaluates to true under $\mathcal{Z}$ and $\mathcal{I}$ if $(s^{\mathcal{I},\mathcal{Z}}, t^{\mathcal{I},\mathcal{Z}}) \in T^\mathcal{I}$. $C(t)$ evaluates to true under $\mathcal{Z}$ and $\mathcal{I}$ if $t^{\mathcal{I},\mathcal{Z}} \in C^\mathcal{I}$. A P2RPQ $q = \exists \vec{x}.\varphi$ is *satisfied* by $\mathcal{I}$ (written: $\mathcal{I} \models q$) if there is an assignment $\mathcal{Z}$ (called *match*) such that φ evaluates to true under $\mathcal{I}$ and $\mathcal{Z}$. A P2RPQ q is *entailed* by a KB $\mathcal{K}$ (written: $\mathcal{K} \models q$) if every model of $\mathcal{K}$ satisfies q. If q uses only signature elements of $\mathcal{K}$, we call it a query *over* $\mathcal{K}$.

Simplifications

A P2RPQ is *simplified* if all its atoms are of the form $T(x,y)$ where x and y are variables and T is built from role names using $\cup$, $\circ$, and *. For any KB $\mathcal{K}$ and P2RPQ q, we can construct in polynomial time a KB $\mathcal{K}'$ and simplified P2RPQ q' such that $\mathcal{K} \models q$ iff $\mathcal{K}' \models q'$. This justifies that from here on, we will focus on simplified queries.

Automata

For each atom $T(x,y)$ of a simplified P2RPQ, T is a regular expression over $\mathbf{N_R}$. Every such expression can be represented by a nondeterministic finite automaton $\mathfrak{A} = (\Sigma, \mathfrak{Q}, \mathfrak{I}, \mathfrak{F}, \mathfrak{T})$ with an alphabet $\Sigma \subseteq \{P, P^- \mid P \in \mathbf{N_R}\}$ of possibly inverted role names, a finite set $\mathfrak{Q}$ of states, initial states $\mathfrak{I} \subseteq \mathfrak{Q}$, final states $\mathfrak{F} \subseteq \mathfrak{Q}$, and transitions $\mathfrak{T} \subseteq \mathfrak{Q} \times \Sigma \times \mathfrak{Q}$.[2] The size of $\mathfrak{A}$ is polynomially bounded by T. In the following, we assume queries in *automaton form*, where the atoms $T(x,y)$ have been replaced by the corresponding $\mathfrak{A}(x,y)$. Given $\mathfrak{A} = (\Sigma, \mathfrak{Q}, \mathfrak{I}, \mathfrak{F}, \mathfrak{T})$, let $\mathfrak{A}^- = (\Sigma^-, \mathfrak{Q}, \mathfrak{F}, \mathfrak{I}, \mathfrak{T}^-)$ with $\Sigma^- = \{P^- \mid P \in \Sigma\}$ be the corresponding reverse automaton[3] with initial and final states swapped and the state transitions flipped: $(\mathfrak{q}', R^-, \mathfrak{q}) \in \mathfrak{T}^-$ for every $(\mathfrak{q}, R, \mathfrak{q}') \in \mathfrak{T}$. Moreover, we obtain the automaton $\mathfrak{A}_{\mathfrak{q},\mathfrak{q}'} = (\Sigma, \mathfrak{Q}, \{\mathfrak{q}\}, \{\mathfrak{q}'\}, \mathfrak{T})$ from $\mathfrak{A}$ by setting the initial state to $\mathfrak{q}$ and the final state to $\mathfrak{q}'$.

3 A Little Bit of Model Theory

The "well-behavedness" of certain sublogics of $\mathcal{ZOIQ}$ can be conveniently characterised in model-theoretic terms.

Definition 1 (quasi-forest model, Calvanese *et al.*, 2007) *Let $\mathcal{K}$ be a KB. A model $\mathcal{I}$ of $\mathcal{K}$ is a* quasi-forest model *if:*

- *the domain $\Delta^\mathcal{I}$ of $\mathcal{I}$ is a forest, i.e., a prefix-closed subset of $Roots \times \mathbb{N}^*$ for some finite set $Roots$,*
- *$Roots = \{o^\mathcal{I} \mid o$ is an individual name in $\mathcal{K}\}$, and*
- *for every $\delta, \delta' \in \Delta^\mathcal{I}$ with $(\delta, \delta') \in P^\mathcal{I}$ for some role name $P \in \mathbf{N_R} \setminus \{\overline{\top}\}$, either (i) $\{\delta, \delta'\} \cap Roots \neq \emptyset$, or (ii) $\delta = \delta'$, or (iii) δ is a child of δ', or (iv) δ' is a child of δ.*

A KB $\mathcal{K}$ has the quasi-forest model property *(short: qfmp) if $\mathcal{K}$ is either unsatisfiable or it has a quasi-forest model. A DL $\mathcal{L}$ has the qfmp if every $\mathcal{L}$ KB $\mathcal{K}$ has the qfmp.*

For such well-behaved KBs, tight and quite decent complexity bounds for satisfiability checking have been established.

[1] Note that $\exists \vec{x}. \bigvee_i \varphi_i \equiv \bigvee_i \exists \vec{x}.\varphi_i$, so the order of disjunction and existential quantification is irrelevant.

[2] The correspondence is such that $(\delta, \delta') \in T^\mathcal{I}$ iff there is a role path $\delta \xrightarrow{P_1} \ldots \xrightarrow{P_k} \delta'$ in $\mathcal{I}$ such that $\mathfrak{A}$ accepts $P_1 \ldots P_k$.

[3] For convenience, we consider $(P^-)^-$ identical to P.

Theorem 1 (Calvanese *et al.*, 2009) *Satisfiability checking of $\mathcal{ZOIQ}$ KBs having the qfmp is* ExpTime*-complete even with binary encoding of number restrictions.*

When it comes to query answering, a useful notion for relating interpretations to each other is via homomorphisms. For two interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ and $\mathcal{J} = (\Delta^\mathcal{J}, \cdot^\mathcal{J})$, a function $h : \Delta^\mathcal{I} \to \Delta^\mathcal{J}$ is a *homomorphism* from $\mathcal{I}$ to $\mathcal{J}$ if for all $\delta, \delta' \in \Delta^\mathcal{I}$ we have $\delta \in A^\mathcal{I} \Rightarrow h(\delta) \in A^\mathcal{J}$ for all concept names A, and $(\delta, \delta') \in P^\mathcal{I} \Rightarrow (h(\delta), h(\delta')) \in P^\mathcal{J}$ for all role names P. If a homomorphism from $\mathcal{I}$ to $\mathcal{J}$ exists, we write $\mathcal{I} \rhd \mathcal{J}$. We note that $\mathcal{I} \models q$ and $\mathcal{I} \rhd \mathcal{J}$ imply $\mathcal{J} \models q$ for every simplified P2RPQ q.

Definition 2 (qfhcp, Bourhis *et al.*, 2014) *A KB $\mathcal{K}$ has the* quasi-forest homomorphism-cover property *(short: qfhcp) if for every model $\mathcal{I}$ of $\mathcal{K}$, there exists a quasi-forest model $\mathcal{J}$ of $\mathcal{K}$ with $\mathcal{J} \rhd \mathcal{I}$. A description logic $\mathcal{L}$ has the qfhcp if every $\mathcal{L}$ KB $\mathcal{K}$ has the qfhcp. We let* tame $\mathcal{ZOIQ}$ *denote the set of all $\mathcal{ZOIQ}$ KBs having the qfhcp.*

Obviously, the qfhcp implies the qfmp. Note that unrestricted $\mathcal{ZOIQ}$ does not even have the qfmp. The most expressive syntactically defined sublogics of tame $\mathcal{ZOIQ}$ known today are $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ [Calvanese *et al.*, 2009; Bonatti *et al.*, 2008].

From the fact that the set of models for any P2RPQ is closed under homomorphisms now follows that for any tame $\mathcal{ZOIQ}$ KB with $\mathcal{K} \not\models q$, there must exist a quasi-forest model $\mathcal{I}$, satisfying $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models q$.

4 Annotating (Partial) Query Matches

In the following, we consider a tame $\mathcal{ZOIQ}$ KB $\mathcal{K}$ and a simplified P2RPQ q over $\mathcal{K}$ in automaton form. We note that q can be rewritten into an equivalent disjunction $\bigvee_{1\leq i\leq n} q_i$ of n (also simplified) C2RPQs, where n may be exponential in the size of q, but every single q_i's size is polynomial. Let $[q]$ denote $\{q_1, \ldots q_n\}$. Then, testing $\mathcal{K} \not\models q$ amounts to determining if $\mathcal{K}$ has a model $\mathcal{I}$ which is a simultaneous countermodel (i.e., $\mathcal{I} \not\models q_i$) for all $q_i \in [q]$.

We now give an intuitive description of our technique to detect (or refute) matches of q. Given some model $\mathcal{I}$ of $\mathcal{K}$, we iteratively, deterministically annotate all its domain elements δ with fresh concept names Q_M that indicate which "parts" M of some query q_i match into $\mathcal{I}$ and how δ participates in these partial matches. To this end, we employ descriptions M of query parts which contain information about (i) the query variables matched (ii) (the existence of) paths realising certain state transitions in the query's automata and (iii) optionally, the "position" of the current δ in the query match by use of a marker $\bullet$ acting like an additional, distinguished query variable. In the annotation process, the Q_Ms will be assigned to domain elements δ based on their local environment and known annotations $Q_{M'}$ for "smaller" partial queries to the same element δ or its direct (role) neighborhood. As an exception, non-localised query matches M not containing $\bullet$ will be "broadcast", i.e., uniformly attached to all domain elements. This way, in the annotation process, Q_Ms for larger and larger partial queries are assigned, until finally (after reaching the unique fixpoint) also the full matches for any q_i are recognised by annotations Q_{q_i}. This process will accurately detect partial and full query matches, if $\mathcal{I}$ is a quasi-forest model (which is sufficient for our purposes as $\mathcal{K}$ has the qfhcp). The annotation process is realised by virtue of a KB $\mathcal{K}_q$, the size of which is exponential in q, but only polynomial in $\mathcal{K}$. In the following, we will stepwise introduce the KB, interleaved with some necessary definitions.

Definition 3 (partial query) *Consider some* C2RPQ $q_i = \{\mathfrak{A}_1(x_1, y_1), \ldots, \mathfrak{A}_m(x_m, y_m)\}$.[4] *A* partial query *for q_i and a set $X \subseteq \{x_1, y_1, \ldots, x_m, y_m\}$ of variables from q_i is a set M consisting of*

- *all atoms $\mathfrak{A}(x, y)$ from q_i for which $\{x, y\} \subseteq X$,*
- *for every $\mathfrak{A}(x, y)$ from q_i with $x \in X$ but $y \notin X$, one of the atoms $\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(x, \bullet)$ or $\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(x, o)$ where q is initial in $\mathfrak{A}$,*
- *for every $\mathfrak{A}(x, y)$ from q_i with $x \notin X$ but $y \in X$, one of the atoms $\mathfrak{A}^-_{\mathsf{q},\mathsf{q}'}(y, \bullet)$ or $\mathfrak{A}^-_{\mathsf{q},\mathsf{q}'}(y, o)$ where q is final in $\mathfrak{A}$,*
- *in case $X = \emptyset$, exactly one atom of the form $\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(\bullet, o)$ or $\mathfrak{A}^-_{\mathsf{q},\mathsf{q}'}(\bullet, o)$ (called* nominal-anchored path*) or $\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(\bullet, \bullet)$ or $\mathfrak{A}^-_{\mathsf{q},\mathsf{q}'}(\bullet, \bullet)$ (called* round trip*) for some $\mathfrak{A}$ from q_i.*

A partial query is called monodic, *if $|X| = 1$. A partial query is called* local *if it contains $\bullet$ and* global *otherwise.*

Nominal-Anchored Paths

First, we want to detect role paths that start in the to-be-annotated individual, end in named individual $o^\mathcal{I}$, and realise state transitions in one of the query's automata. Let o be any individual name in $\mathcal{K}$, let $\mathfrak{A}^\pm$ be either $\mathfrak{A}$ or $\mathfrak{A}^-$ for any automaton $\mathfrak{A}$ occurring in q with states $\mathsf{q}, \mathsf{q}', \mathsf{q}''$. Then let $\mathcal{K}_q$ contain the axiom

$$Q_{\{\mathfrak{A}^\pm_{\mathsf{q},\mathsf{q}}(\bullet,o)\}}(o) \tag{1}$$

and whenever $\mathfrak{A}^\pm$ has a transition $(\mathsf{q}, R, \mathsf{q}')$:

$$\exists R.Q_{\{\mathfrak{A}^\pm_{\mathsf{q}',\mathsf{q}''}(\bullet,o)\}} \sqsubseteq Q_{\{\mathfrak{A}^\pm_{\mathsf{q},\mathsf{q}''}(\bullet,o)\}} \tag{2}$$

Round Trips

Next, we are concerned with paths which start and end in the to-be-annotated individual. Assuming an automaton $\mathfrak{A}$ with states $\mathsf{q}, \mathsf{q}', \mathsf{q}'', \mathsf{q}'''$ and transition set $\mathfrak{T}$ as well as an individual name o, we add the axioms:

$$\top \sqsubseteq Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}}(\bullet,\bullet)\}} \tag{3}$$

$$Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)\}} \sqcap Q_{\{\mathfrak{A}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)\}} \sqsubseteq Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}''}(\bullet,\bullet)\}} \tag{4}$$

$$Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(\bullet,o)\}} \sqcap Q_{\{\mathfrak{A}^-_{\mathsf{q}'',\mathsf{q}'}(\bullet,o)\}} \sqsubseteq Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}''}(\bullet,\bullet)\}} \tag{5}$$

for $(\mathsf{q}, R, \mathsf{q}') \in \mathfrak{T}$:

$$\exists R.\mathsf{Self} \sqsubseteq Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)\}} \tag{6}$$

for $\{(\mathsf{q}, R, \mathsf{q}'), (\mathsf{q}'', R', \mathsf{q}''')\} \subseteq \mathfrak{T}$:

$$\exists(R \cap R'^-).Q_{\{\mathfrak{A}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)\}} \sqsubseteq Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'''}(\bullet,\bullet)\}} \tag{7}$$

Initialising monodic partial queries

Next we determine domain elements to which separate query variables could possibly be mapped in a match.

Definition 4 *We call a monodic partial query M* original, *if $\mathsf{q} = \mathsf{q}'$ for every $\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(x, \bullet) \in M$ and every $\mathfrak{A}^-_{\mathsf{q},\mathsf{q}'}(x, \bullet) \in M$. For M an original monodic partial query for q and $\{x\}$, the set $Prec_M$ of* precondition concepts *consists of:*

[4] In cases, we may equate a C2RPQ q_i with the set of its atoms.

- $Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(\bullet,o)\}}$ *for each atom* $\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(x,o)$ *in* M,
- $Q_{\{\mathfrak{A}^-_{\mathsf{q},\mathsf{q}'}(\bullet,o)\}}$ *for each atom* $\mathfrak{A}^-_{\mathsf{q},\mathsf{q}'}(x,o)$ *in* M,
- $\bigsqcup_{\mathsf{q}\text{ initial},\ \mathsf{q}'\text{ final}} Q_{\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(\bullet,\bullet)\}}$ *for each atom* $\mathfrak{A}(x,x)$ *in* M.

Now we can initialise all original monodic partial queries M by adding the following axioms to $\mathcal{K}_q$:

$$\bigsqcap Prec_M \sqsubseteq Q_M \tag{8}$$

Updating partial queries

Partial queries can be updated by roundtrips: Let M be a partial query containing some $\mathfrak{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)$. Then we realise the update by adding the following axioms to $\mathcal{K}_q$:

$$Q_M \sqcap Q_{\{\mathfrak{A}^{\pm}_{\mathsf{q}',\mathsf{q}''}(\bullet,\bullet)\}} \sqsubseteq Q_{M\setminus\{\mathfrak{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)\}\cup\{\mathfrak{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(x,\bullet)\}} \tag{9}$$

Furthermore, a partial query can "progress" when making a "step" in the model, moving from the considered δ to some role neighbour. In such a step, all the "unready" paths (corresponding to atoms with $\bullet$) must be updated in a synchronous manner. Given a partial query M for q_i and nonempty X as well as a set $\{R_1,\ldots,R_k\}$ of (possibly inverted) role names, assume M' can be obtained from M by replacing each atom of the form $\mathfrak{A}^{\pm}_{\mathsf{q},\mathsf{q}'}(x,\bullet)$ by an atom $\mathfrak{A}^{\pm}_{\mathsf{q},\mathsf{q}''}(x,\bullet)$ such that $\mathfrak{A}^{\pm}$ has a transition $(\mathsf{q}',R,\mathsf{q}'')$ for any $R\in\{R_1,\ldots,R_k\}$. Then add the following axiom to the KB $\mathcal{K}_q$:

$$\exists(R_1^-\cap R_2^-\cap\ldots\cap R_m^-).Q_M \sqsubseteq Q_{M'} \tag{10}$$

Joining partial queries

When two partial queries corresponding to the same query q_i "meet" in a domain element δ, they can – under certain circumstances – be "glued together" into a "bigger" partial query of q_i.

Definition 5 (joinable, join) *Let M_1 be a partial query for q_i and X_1 and let M_2 be a partial query for q_i and X_2. We call M_1 and M_2* joinable *if*

- $X_1\neq\emptyset$, $X_2\neq\emptyset$, $X_1\cap X_2=\emptyset$, *and*
- *for each $\mathfrak{A}(x,y)\in q_i\setminus(M_1\cup M_2)$ with $\{x,y\}\subseteq X_1\cup X_2$, there are states $\mathsf{q},\mathsf{q}',\mathsf{q}''$ of $\mathfrak{A}$ with q initial and q'' final, such that either $\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(x,\bullet),\mathfrak{A}^-_{\mathsf{q}'',\mathsf{q}'}(y,\bullet)\}\in M_1\cup M_2$ or $\{\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(x,o),\mathfrak{A}^-_{\mathsf{q}'',\mathsf{q}'}(y,o)\}\in M_1\cup M_2$ for some $o\in\mathbf{N_I}$.*

For joinable M_1 and M_2, their join*, denoted $M_1\bowtie M_2$, is the partial query obtained from $M_1\cup M_2$ by replacing any pair of atoms $\mathfrak{A}_{\mathsf{q},\mathsf{q}'}(x,*)$ and $\mathfrak{A}^-_{\mathsf{q}'',\mathsf{q}'}(y,*)$ by $\mathfrak{A}(x,y)$, where $*$ is either an individual name o or $\bullet$.*

We implement the join operation for every pair M_1, M_2 of joinable partial queries for a $q_i\in[q]$ by extending $\mathcal{K}_q$ with:

$$Q_{M_1}\sqcap Q_{M_2}\sqsubseteq Q_{M_1\bowtie M_2} \tag{11}$$

Broadcasting global partial queries

Whenever a partial query M does not have any occurrences of $\bullet$, which would tie it to a specific element, it will be "broadcast" to all domain elements:

$$\exists\overline{\top}.Q_M\sqsubseteq Q_M \tag{12}$$

This concludes the definition of $\mathcal{K}_q$. Revisiting our initial intuition of $\mathcal{K}_q$'s purpose of deterministically annotating a model $\mathcal{I}$ of $\mathcal{K}$, the following lemma formalises this by singling out one unique annotated model $\mathcal{I}^*$ for every $\mathcal{I}$.

Lemma 2 *Let $\mathcal{K}$ be a $\mathcal{ZOIQ}$ KB and q a simplified* P2RPQ. *For every model $\mathcal{I}$ of $\mathcal{K}$ there exists a unique model $\mathcal{I}^*$ of $\mathcal{K}\cup\mathcal{K}_q$ which extends $\mathcal{I}$ and satisfies $Q_M^{\mathcal{I}^*}\subseteq Q_M^{\mathcal{J}}$ for every Q_M in $\mathcal{K}_q$ and for every model $\mathcal{J}$ of $\mathcal{K}\cup\mathcal{K}_q$ that extends $\mathcal{I}$.*

We will call models $\mathcal{I}^*$ from Lemma 2 *Q-minimal*. Note that for every model $\mathcal{I}$ of $\mathcal{K}$, any concept membership $\delta\in Q_M^{\mathcal{I}^*}$ holding in the corresponding Q-minimal model $\mathcal{I}^*$ can be derived from concept and role memberships in $\mathcal{I}$ through a finite sequence of "applications" of axioms from $\mathcal{K}_q$.[5]

5 Reduction to Satisfiability

Now we establish technical results relating the presence of Q_M-annotations in models and the actual semantic satisfaction of the corresponding partial query M. Note that any partial query M can be seen as a (set representation of a) C2RPQ in automaton form, assuming $\bullet$ is just an ordinary variable.

Definition 6 ((tight) satisfaction of a partial query) *Given an interpretation $\mathcal{I}$, a domain element $\delta\in\Delta^{\mathcal{I}}$, and a partial query M, we say M is* satisfied *or* holds *in δ, written $\mathcal{I},\delta\models M$, if there is a match $\mathcal{Z}$ for M in $\mathcal{I}$ with $\mathcal{Z}(\bullet)=\delta$. M is* satisfied *or* holds tightly *in δ, written $\mathcal{I},\delta\models\!\!\!\models M$, if $\mathcal{Z}$ is such that every $\mathfrak{A}^{\pm}_{\mathsf{q}_1,\mathsf{q}_2}(x,\bullet)\in M$ is realized by a path $\mathcal{Z}(x)\rightsquigarrow\delta$ not containing $o^{\mathcal{I}}$ for any individual name o.*

Note that $\models$ and $\models\!\!\!\models$ coincide whenever M is global or does not contain variables. Note also that, as a consequence of this definition, if M is global, then Q_M holds (tightly) everywhere or nowhere throughout the domain. With this notion in place, the next two lemmas can be seen as soundness and completeness results regarding the deduction calculus for (partial) query matches embodied by $\mathcal{K}_q$.

Lemma 3 (soundness) *Let $\mathcal{I}$ be a Q-minimal model of $\mathcal{K}\cup\mathcal{K}_q$ and let $\delta\in\Delta^{\mathcal{I}}$. Let M be any partial query for any* C2RPQ *$q_i\in[q]$. Then $\delta\in Q_M^{\mathcal{I}}$ implies $\mathcal{I},\delta\models M$.*

Lemma 4 (completeness) *Let $\mathcal{I}$ be a quasi-forest model of $\mathcal{K}\cup\mathcal{K}_q$ and let $\delta\in\Delta^{\mathcal{I}}$. Let M be any partial query for any* C2RPQ *$q_i\in[q]$. Then $\mathcal{I},\delta\models\!\!\!\models M$ implies $\delta\in Q_M^{\mathcal{I}}$.*

Soundness is proven by induction over the length of the derivation for $\delta\in Q_M^{\mathcal{I}}$. Completeness is shown by induction over the *spread* of the match for M, i.e., the number of variables (excluding $\bullet$) plus the sum of the lengths of all paths realising the query atoms.

Thanks to these correspondences, we can essentially rule out models with matches of any $q_1,\ldots,q_n$ by forcing the extensions of $Q_{q_1},\ldots,Q_{q_n}$ to be empty. Therefore, let

$$\mathcal{K}_{\neg q}=\mathcal{K}\cup\mathcal{K}_q\cup\{Q_{q_i}\sqsubseteq\bot\mid q_i\in[q]\}. \tag{13}$$

We establish some syntactic and semantic properties of $\mathcal{K}_{\neg q}$: Some easy calculations yield size bounds for $\mathcal{K}_{\neg q}$, tameness is not affected by the reduction, and unsatisfiability of $\mathcal{K}_{\neg q}$ indeed coincides with query entailment.

Fact 5 (size of $\mathcal{K}_{\neg q}$) *The size of $\mathcal{K}_{\neg q}$ is single exponential in the total size of $\mathcal{K}$ and q. It is polynomial, if the number of atoms in q is bounded by a constant.*

[5] To see this, it may help to realise that all axioms of $\mathcal{K}_q$ can be easily expressed in monadic Datalog.

Lemma 6 *If $\mathcal{K}$ is in tame $\mathcal{ZOIQ}$ then so is $\mathcal{K}_{\neg q}$.*

Proof. First note that $\mathcal{K}_{\neg q}$ is in $\mathcal{ZOIQ}$. For tameness, consider a model $\mathcal{I}$ of $\mathcal{K}_{\neg q}$. As $\mathcal{I}$ is a model of $\mathcal{K}$ and the latter is tame, there exists a quasi-forest model $\mathcal{J}$ of $\mathcal{K}$ and a homomorphism $h : \mathcal{J} \to \mathcal{I}$. Now, we can construct a quasi-forest model $\mathcal{J}'$ of $\mathcal{K}_{\neg q}$ by extending $\mathcal{J}$, letting $\delta \in Q_M^{\mathcal{J}'}$ if $h(\delta) \in Q_M^{\mathcal{I}}$ for all Q_M occurring in $\mathcal{K}$. □

Theorem 7 *Let $\mathcal{K}$ be a tame $\mathcal{ZOIQ}$ KB and let q be a simplified* P2RPQ *over $\mathcal{K}$. Then $\mathcal{K} \models q$ if and only if $\mathcal{K}_{\neg q}$ is unsatisfiable.*

Proof. For convenience, we show the equivalent statement: $\mathcal{K} \not\models q$ if and only if $\mathcal{K}_{\neg q}$ satisfiable.
"only if": Assume $\mathcal{K} \not\models q$, that is there exists an $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models q$. Consider $\mathcal{I}^*$, the Q-minimal model of $\mathcal{K} \cup \mathcal{K}_q$ extending $\mathcal{I}$. If some $Q_{q_i}^{\mathcal{I}^*}$ were nonempty, then there would be a query match of q into $\mathcal{I}$ by Lemma 3, contradicting our assumption. Therefore, $\mathcal{I}^* \models Q_{q_i} \sqsubseteq \bot$ for all $q_i \in [q]$ and hence, $\mathcal{I}^* \models \mathcal{K}_{\neg q}$.
"if": Assume $\mathcal{K}_{\neg q}$ is satisfiable and hence has a model $\mathcal{I}$. As $\mathcal{K}_{\neg q}$ has the qfmp thanks to Lemma 6, we can assume $\mathcal{I}$ to be a quasi-forest model. $\mathcal{I} \models Q_{q_i} \sqsubseteq \bot$ implies emptiness of $Q_{q_i}^{\mathcal{I}}$ for every $q_i \in [q]$. If $\mathcal{I} \models q$ held true, witnessed by $\mathcal{I} \models q_i$ for some $q_i \in [q]$, Lemma 4 would require $\delta \in Q_{q_i}^{\mathcal{I}}$ for all $\delta \in \Delta^{\mathcal{I}}$, leading to a contradiction. Hence $\mathcal{I} \not\models q$. On the other hand, $\mathcal{I} \models \mathcal{K}_{\neg q}$ implies $\mathcal{I} \models \mathcal{K}$, therefore $\mathcal{K} \not\models q$. □

Based on this theorem, we can now prove our central result:

Theorem 8 *The problem of checking entailment of simplified* P2RPQ*s from tame $\mathcal{ZOIQ}$ KBs with binary number encoding is* 2EXPTIME*-complete. It is* EXPTIME*-complete if the number of atoms in the query is bounded.*

Proof. By Theorem 7, entailment of a simplified C2RPQ q from a tame $\mathcal{ZOIQ}$ KB $\mathcal{K}$ can be reduced to checking unsatisfiability of $\mathcal{K}_{\neg q}$, which can be computed in output-polynomial time and the size of which is exponential (polynomial, if number of atoms in q is bounded). By Theorem 1, (un)satisfiability of $\mathcal{K}_{\neg q}$ can be checked in EXPTIME in the size of $\mathcal{K}_{\neg q}$, hence we obtain the desired 2EXPTIME and EXPTIME upper bounds. The matching lower bounds are well-known even for much weaker logics [Ortiz and Simkus, 2012]. □

6 Querying $\mathcal{GC}^2$ Databases

We now consider the problem of P2RPQ entailment from $\mathcal{GC}^2$ databases, i.e., sets of unary and binary ground facts in the presence of a sentence in the guarded two-variable fragment with counting quantifiers as defined by Pratt-Hartmann (2007). We will show that there is a poly-time query-entailment-preserving transformation from $\mathcal{GC}^2$ databases to $\mathcal{ZIQ}$ KBs. This allows us to transfer the results from Theorem 8. Matching lower bounds follow from known results for much weaker logics and query formalisms (such as CQs over $\mathcal{ALCI}$ [Lutz, 2008]).

Definition 7 ($\mathcal{GC}^2$**; Pratt-Hartmann, 2007)** *For a signature $\mathbf{S}$ of nullary ($\mathbf{N_0}$), unary ($\mathbf{N_C}$), and binary ($\mathbf{N_R} \cup \{\approx\}$) predicates,*[6] *let $\mathcal{GC}^2$ be the smallest set of formulae that contains all atoms over $\mathbf{S}$ using only variables from $\{x, y\}$, that is closed under Boolean combinations, and that contains*

- *all formulae $\exists u.\varphi$ and $\forall u.\varphi$ with $u \in \{x, y\}$ whenever φ is a $\mathcal{GC}^2$ formula with at most one free variable, and*
- *all formulae $\forall u.(\gamma \to \varphi)$ as well as $\circlearrowright u.(\gamma \wedge \varphi)$ and $\circlearrowright u.(\gamma)$ where γ is a binary atom (called "guard") containing both x and y, φ is a $\mathcal{GC}^2$ formula, and $\circlearrowright$ is any of $\exists$, $\exists_{\leqslant n}$, $\exists_{=n}$, or $\exists_{\geqslant n}$.*

A $\mathcal{GC}^2$ database DB *is a theory $\{\Phi\} \cup \mathcal{A}$ consisting of a $\mathcal{GC}^2$ sentence Φ and a finite set $\mathcal{A}$ of unary and binary ground facts over constants from $\mathbf{N_I}$.*[7] *We denote by $\mathcal{GC}^2_{\mathrm{qef}}$ the set of $\mathcal{GC}^2$ formulae which are quantifier-free and do not use $\approx$.*

A P2RPQ over a $\mathcal{GC}^2$ database is a P2RPQ where for every atom of the form $C(t)$ holds $C \in \mathbf{N_C}$ and for every atom of the form $T(s, t)$, every subexpression of the form $id(C)$ satisfies $C \in \mathbf{N_C}$. P2RPQ entailment from $\mathcal{GC}^2$ databases is defined in the same way as for DLs.

For convenience, we make use of a special normal form for $\mathcal{GC}^2$ introduced in the following.

Lemma 9 (normal form; Pratt-Hartmann, 2009) *For any $\mathcal{GC}^2$ sentence Φ, one can compute in polynomial time a $\mathcal{GC}^2$ sentence* $\mathsf{NF}(\Phi)$ *of the form*

$$\begin{aligned} \forall x\alpha \wedge \textstyle\bigwedge_{1\leq h\leq \ell} \forall x\forall y\, (E_h(x,y) \to \beta_h \vee x \approx y)) \\ \wedge \textstyle\bigwedge_{1\leq i\leq m} \forall x \exists_{=n_i} y\, (F_i(x,y) \wedge x \not\approx y)\,, \end{aligned} \tag{14}$$

with $\alpha \in \mathcal{GC}^2_{\mathrm{qef}}$ not using y, $\{E_1, \ldots, E_\ell, F_1, \ldots, F_m\} \subseteq \mathbf{N_R}$, $\{\beta_1, \ldots, \beta_\ell\} \subseteq \mathcal{GC}^2_{\mathrm{qef}}$, and $\{n_1, \ldots, n_m\} \subseteq \mathbb{N}$, such that (i) $\mathsf{NF}(\Phi) \models \Phi$ and (ii) every model of Φ with at least $\max_h n_h$ elements can be extended to a model of $\mathsf{NF}(\Phi)$.

Given a $\mathcal{GC}^2$ database DB $= \{\Phi\} \cup \mathcal{A}$, we let $\mathsf{NF}(\mathrm{DB}) = \{\mathsf{NF}(\Phi)\} \cup \mathcal{A}$. Thanks to the previous lemma and the known property that models of $\mathcal{GC}^2$ sentences are closed under disjoint self-union, the normal form can be shown to be query-entailment-preserving.

Lemma 10 *For every* P2RPQ *q over a $\mathcal{GC}^2$ database* DB *holds* DB $\models q$ *if and only if* $\mathsf{NF}(\mathrm{DB}) \models q$*.*

We next provide a query-entailment-preserving polynomial translation of $\mathcal{GC}^2$ databases into $\mathcal{ZIQ}$ KBs.

Definition 8 *Let Ψ be a $\mathcal{GC}^2$ sentence in normal form as in Lemma 9. We define the $\mathcal{ZIQ}$ TBox $\mathcal{T}_\Psi$ by*

$$\mathcal{T}_\Psi = \mathcal{T}_\approx \cup \mathcal{T}_{\mathrm{ps}} \cup \mathcal{T}_\alpha \cup \bigcup_{1\leq h\leq l} \mathcal{T}_{E_h,\beta_h} \cup \bigcup_{1\leq i\leq m} \mathcal{T}_{F_i,n_i}, \tag{15}$$

where (introducing fresh concept names $Prop_P$ and role names $R_\approx$ and $R_{E_h,at}$):

[6] As unary and binary predicates correspond to concept and role names, respectively, we use the same set symbols to denote them.

[7] Note that ABoxes, as defined before, are syntactically and semantically identical to finite sets of unary and binary ground facts, so we consider the notions the same and use the same symbol $\mathcal{A}$. Note also that by definition, Φ does not contain constants.

- $\mathcal{T}_{\approx} = \{\top \sqsubseteq \leqslant 1 R_{\approx}.\top,\ \top \sqsubseteq \exists R_{\approx}.\mathsf{Self}\}$
- $\mathcal{T}_{\mathrm{ps}} = \{Prop_P \equiv \exists \overline{\top}.Prop_P \mid P \in \mathbf{N_0}\}$
- $\mathcal{T}_{\alpha} = \{\top \sqsubseteq \mathsf{tr}(\alpha)\}$, *where*

$$\begin{array}{rlrl} \mathsf{tr}(P) = Prop_P & & \mathsf{tr}(\neg\varphi) = \neg\mathsf{tr}(\varphi) \\ \mathsf{tr}(P(x)) = P & & \mathsf{tr}(\varphi \wedge \psi) = \mathsf{tr}(\varphi) \sqcap \mathsf{tr}(\psi) \\ \mathsf{tr}(P(x,x)) = \exists P.\mathsf{Self} & & \mathsf{tr}(\varphi \vee \psi) = \mathsf{tr}(\varphi) \sqcup \mathsf{tr}(\psi) \end{array}$$

- $\mathcal{T}_{E_h,\beta_h} = \bigcup_{at \in \mathsf{atoms}(\beta_h)} \mathsf{ax}_{E_h}(at)$
  $\cup \{\top \sqsubseteq \forall (R_{E_h,at} \setminus E_h).\bot \mid at \in \mathsf{atoms}(\beta_h)\}$
  $\cup \{\top \sqsubseteq \forall (E_h \setminus (R_{\approx} \cup \mathsf{tr}_{E_h}(\beta_h))).\bot\}$

  where ax_{E_h} *maps atoms in* β_h *to sets of GCIs as follows:*

$$\begin{array}{lll} P & \mapsto \{ & \exists R_{E_h,P}.\top \sqsubseteq Prop_P, \\ & & \exists (E_h \setminus R_{E_h,P}).\top \sqsubseteq \neg Prop_P \ \} \\ P(x) & \mapsto \{ & \exists R_{E_h,P(x)}.\top \sqsubseteq P, \\ & & \exists (E_h \setminus R_{E_h,P(x)}).\top \sqsubseteq \neg P \ \} \\ P(x,x) & \mapsto \{ & \exists R_{E_h,P(x,x)}.\top \sqsubseteq \exists P.\mathsf{Self}, \\ & & \exists (E_h \setminus R_{E_h,P(x,x)}).\top \sqsubseteq \neg\exists P.\mathsf{Self} \} \\ P(y) & \mapsto \{ & \exists R^-_{E_h,P(y)}.\top \sqsubseteq P, \\ & & \exists (E^-_h \setminus R^-_{E_h,P(y)}).\top \sqsubseteq \neg P \ \} \\ P(y,y) & \mapsto \{ & \exists R^-_{E_h,P(y,y)}.\top \sqsubseteq \exists P.\mathsf{Self}, \\ & & \exists (E^-_h \setminus R^-_{E_h,P(y,y)}).\top \sqsubseteq \neg\exists P.\mathsf{Self} \} \\ P(x,y) & \mapsto \{ & \exists (R_{E_h,P(x,y)} \setminus P).\top \sqsubseteq \bot, \\ & & \exists (E_h \cap P \setminus R_{E_h,P(x,y)}).\top \sqsubseteq \bot \ \} \\ P(y,x) & \mapsto \{ & \exists (R^-_{E_h,P(y,x)} \setminus P).\top \sqsubseteq \bot, \\ & & \exists (E^-_h \cap P \setminus R^-_{E_h,P(y,x)}).\top \sqsubseteq \bot \ \} \end{array}$$

  and tr_{E_h} *maps* $\mathcal{GC}^2_{\mathrm{qef}}$ *formulae to simple roles:*

$$\begin{array}{r} \mathsf{tr}_{E_h}(at) = R_{E_h,at} \\ \mathsf{tr}_{E_h}(\neg\rho) = E_h \setminus \mathsf{tr}(\rho) \\ \mathsf{tr}_{E_h}(\rho \wedge \rho') = \mathsf{tr}_{E_h}(\rho) \cap \mathsf{tr}_{E_h}(\rho') \\ \mathsf{tr}_{E_h}(\rho \vee \rho') = \mathsf{tr}_{E_h}(\rho) \cup \mathsf{tr}_{E_h}(\rho') \end{array}$$

- $\mathcal{T}_{F_i,n_i} = \{\ \top \sqsubseteq\ \leqslant n_i(F_i \setminus R_{\approx}).\top \sqcap \geqslant n_i(F_i \setminus R_{\approx}).\top\ \}$

Given a $\mathcal{GC}^2$ *database* $\mathrm{DB} = \{\Phi\} \cup \mathcal{A}$, *we let* $\mathcal{K}_{\mathrm{DB}}$ *denote the* $\mathcal{ZIQ}$ *KB* $(\mathcal{T}_{\mathsf{NF}(\Phi)}, \mathcal{A})$.

Intuitively, $\mathcal{T}_{\approx}$ axiomatises equality, $\mathcal{T}_{\mathrm{ps}}$ introduces synchronised concept names $Prop_P$ for every propositional symbol P from DB, since $\mathcal{ZIQ}$ does not allow for nullary predicates. $\mathcal{T}_{\alpha}$, $\mathcal{T}_{E_h,\beta_h}$, and $\mathcal{T}_{F_i,n_i}$ implement the conjuncts from $\mathsf{NF}(\Phi)$.

Lemma 11 *Let* $\mathrm{DB} = \{\Phi\} \cup \mathcal{A}$ *be a* $\mathcal{GC}^2$ *database. Then the following hold:*

1. $\mathcal{K}_{\mathrm{DB}}$ *can be computed in polytime wrt. the size of* DB.
2. $\mathcal{K}_{\mathrm{DB}}$ *has the qfhcp.*
3. *Any model of* $\mathcal{K}_{\mathrm{DB}}$ *can be extended to one of* $\mathsf{NF}(\mathrm{DB})$.
4. *Any model of* $\mathsf{NF}(\mathrm{DB})$ *can be extended to one of* $\mathcal{K}_{\mathrm{DB}}$.
5. *For any* P2RPQ q *over* DB *holds* $\mathrm{DB} \models q$ *iff* $\mathcal{K}_{\mathrm{DB}} \models q$.

Proof. Point 1 is obvious from the given translation. Point 2 is a direct consequence from the fact that $\mathcal{K}_{\mathrm{DB}}$ is a $\mathcal{ZIQ}$ KB. For Point 3, we have to extend the model $\mathcal{I}$ by interpretations for all $P \in \mathbf{N_0}$: we map P to true exactly if $Prop_P = \Delta^{\mathcal{I}}$. For Point 4, we have to extend the model $\mathcal{I}$ by interpretations for all $Prop_P$, $R_{\approx}$, and $R_{E_h,at}$: we let $Prop_P^{\mathcal{I}} = \Delta^{\mathcal{I}}$ if P holds true in $\mathcal{I}$ and $Prop_P^{\mathcal{I}} = \emptyset$ otherwise; we let $R_{\approx}^{\mathcal{I}} = \{(\delta,\delta) \mid \delta \in \Delta\}$; and we let $R_{E_h,at}^{\mathcal{I}} = \{(\mathcal{Z}(x), \mathcal{Z}(y)) \mid \mathcal{I}, \mathcal{Z} \models E_h(x,y) \wedge at\}$. Point 5 can be shown in two steps: by Lemma 10, the queries entailed by DB and by $\mathsf{NF}(\mathrm{DB})$ coincide. Then, as a consequence of the fact that by Point 3 and Point 4, the model sets of $\mathsf{NF}(\mathrm{DB})$ and $\mathcal{K}_{\mathrm{DB}}$ coincide on all unary and binary predicates occurring in DB, hence also the entailed queries coincide. ☐

Theorem 12 P2RPQ *entailment from* $\mathcal{GC}^2$ *databases is* 2EXPTIME*-complete. It is* EXPTIME*-complete if the number of query atoms is bounded by a constant.*

Proof. Given a $\mathcal{GC}^2$ database DB and a P2RPQ q over DB, we can compute $\mathcal{K}_{\mathrm{DB}}$ in polynomial time, and it is of polynomial size by Lemma 11, Point 1. Moreover, $\mathrm{DB} \models q$ can be checked by testing $\mathcal{K}_{\mathrm{DB}} \models q$ by Lemma 11, Point 5, which in turn can be checked in doubly exponential time (singly exponential if q is bounded) by Theorem 8, since $\mathcal{K}_{\mathrm{DB}}$ has the qfhcp by Lemma 11, Point 2. ☐

7 Conclusion

We have established tight complexity bounds for expressive querying in very expressive DLs under the assumption of succinct (i.e. binary) encoding of number restrictions. Arguing along the lines of Calvanese *et al.* (2009), we can leverage our findings to strengthen their results on query containment as well as the $\mathcal{SR}$ family as follows:[8]

Theorem 13 (query containment) *Testing query entailment* $\mathcal{K} \models q' \subseteq q$ *is in* 2EXPTIME *with respect to the total size of* q', q, *and* $\mathcal{K}$ *(with binary encoding of number restrictions) if (i)* $\mathcal{K}$ *is in* $\mathcal{ZOQ}$ *or* $\mathcal{ZOI}$ *and* q' *and* q *are* P2RPQ*s over* $\mathcal{K}$, *or (ii)* $\mathcal{K}$ *is in* $\mathcal{ZIQ}$, q' *is a conjunctive query and* q *is a* P2RPQ *over* $\mathcal{K}$. *The complexity drops to* EXPTIME *if the number of atoms occurring in* q *is bounded by a constant.*

Theorem 14 ($\mathcal{SR}$ family) *Deciding* $\mathcal{K} \models q$ *and* $\mathcal{K} \models q' \subseteq q$ *is in* 3EXPTIME *in the total size of* q', q, $\mathcal{K}$ *(with binary encoding of numbers) – and in* 2EXPTIME *if* $\mathcal{K}$*'s RBox is given by defining regular expressions for the non-simple roles – if (i)* $\mathcal{K}$ *is in* $\mathcal{SROQ}$ *or* $\mathcal{SROI}$ *and* q', q *are* P2RPQ*s over* $\mathcal{K}$, *or (ii)* $\mathcal{K}$ *is in* $\mathcal{SRIQ}$, q' *is a CQ and* q *is a* P2RPQ *over* $\mathcal{K}$. *The complexities are in* 2EXPTIME *and in* EXPTIME*, respectively, if the number of atoms in* q *is bounded by a constant.*

There are plenty of open questions left for future work. First, the results established here hold for arbitrary models, however, so far, very little is known about finite (model) query answering in DLs from the $\mathcal{Z}$ family. Second, while the extension of our query formalism by nestings in the sense of Bienvenu *et al.* (2014) seems straightforward without impacting complexities, our technique seems not readily extendable to capture more elaborate query languages [Rudolph and Krötzsch, 2013; Bourhis *et al.*, 2015; Reutter *et al.*, 2017].

[8]For space reasons, we have to assume the reader to be familiar with the notions used in these theorems.

References

[Bienvenu *et al.*, 2014] Meghyn Bienvenu, Diego Calvanese, Magdalena Ortiz, and Mantas Simkus. Nested regular path queries in description logics. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proc. 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'14)*, pages 218–227. AAAI Press, 2014.

[Bonatti *et al.*, 2008] Piero A. Bonatti, Carsten Lutz, Aniello Murano, and Moshe Y. Vardi. The complexity of enriched mu-calculi. *Logical Methods in Computer Science*, 4(3:11):1–27, 2008.

[Bourhis *et al.*, 2014] Pierre Bourhis, Markus Krötzsch, and Sebastian Rudolph. How to best nest regular path queries. In *Proc. 27th Int. Workshop on Description Logics (DL'14)*, volume 1193 of *CEUR Workshop Proceedings*, pages 404–415. CEUR-WS.org, 2014.

[Bourhis *et al.*, 2015] Pierre Bourhis, Markus Krötzsch, and Sebastian Rudolph. Reasonable highly expressive query languages. In Qiang Yang and Michael Wooldridge, editors, *Proc. 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15)*, pages 2826–2832. AAAI Press, 2015.

[Calvanese *et al.*, 2007] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proc. 22nd AAAI Conf. on Artificial Intelligence (AAAI'07)*, pages 391–396. AAAI Press, 2007.

[Calvanese *et al.*, 2009] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In Craig Boutilier, editor, *Proc. 21st Int. Joint Conf. on Artificial Intelligence (IJCAI'09)*, pages 714–720. IJCAI, 2009.

[Lutz, 2008] Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Proc. 4th Int. Joint Conf. on Automated Reasoning (IJCAR'08)*, number 5195 in LNAI, pages 179–193. Springer, 2008.

[Ortiz and Simkus, 2012] Magdalena Ortiz and Mantas Simkus. Reasoning and query answering in description logics. In Thomas Eiter and Thomas Krennwallner, editors, *Proc. 8th Int. Reasoning Web Summer School (RW'12)*, volume 7487 of *Lecture Notes in Computer Science*, pages 1–53. Springer, 2012.

[Ortiz *et al.*, 2010] Magdalena Ortiz, Sebastian Rudolph, and Mantas Simkus. Query answering is undecidable in DLs with regular expressions, inverses, nominals, and counting. INFSYS Research Report 1843-10-03, TU Vienna, 2010.

[Pratt-Hartmann, 2007] Ian Pratt-Hartmann. Complexity of the guarded two-variable fragment with counting quantifiers. *J. Log. Comput.*, 17(1):133–155, 2007.

[Pratt-Hartmann, 2009] Ian Pratt-Hartmann. Data-complexity of the two-variable fragment with counting quantifiers. *Inf. Comput.*, 207(8):867–888, 2009.

[Reutter *et al.*, 2017] Juan L. Reutter, Miguel Romero, and Moshe Y. Vardi. Regular queries on graph databases. *Theor. Comp. Sys.*, 61(1):31–83, July 2017.

[Rudolph and Krötzsch, 2013] Sebastian Rudolph and Markus Krötzsch. Flag & check: Data access with monadically defined queries. In Richard Hull and Wenfei Fan, editors, *Proc. 32nd Symposium on Principles of Database Systems (PODS'13)*, pages 151–162. ACM, 2013.

[Rudolph, 2016] Sebastian Rudolph. Undecidability Results for Database-Inspired Reasoning Problems in Very Expressive Description Logics. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Proc. 15th Int. Conf. on Knowledge Representation and Reasoning (KR'16)*, pages 247–257. AAAI Press, 2016.