Query Answering in the Horn Fragments of the Description Logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$*

Magdalena Ortiz† and **Sebastian Rudolph**‡ and **Mantas Šimkus**†

†Vienna University of Technology, Austria
{ortiz@kr,simkus@dbai}.tuwien.ac.at

‡Karlsruhe Institute of Technology, Germany
rudolph@kit.edu

Abstract

The high computational complexity of the expressive Description Logics (DLs) that underlie the OWL standard has motivated the study of their Horn fragments, which are usually tractable in data complexity and can also have lower combined complexity, particularly for query answering. In this paper we provide algorithms for answering conjunctive 2-way regular path queries (2CRPQs), a nontrivial generalization of plain conjunctive queries, in the Horn fragments of the DLs $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ underlying OWL 1 and OWL 2. We show that the combined complexity of the problem is ExpTime-complete for Horn-$\mathcal{SHOIQ}$ and 2ExpTime-complete for the more expressive Horn-$\mathcal{SROIQ}$, but is PTime-complete in data complexity for both. In contrast, even decidability of plain conjunctive queries is still open for full $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. These are the first completeness results for query answering in DLs with inverses, nominals, and counting, and show that for the considered logics the problem is not more expensive than standard reasoning.

1 Introduction

Description Logics (DLs), a family of languages for Knowledge Representation and Reasoning, are applied in many fields and are particularly popular as *ontology languages* for conceptually describing application domains. In fact, the Web Ontology Languages (OWL) proposed as a standard for writing ontologies on the Web are based on expressive DLs: OWL-DL (a.k.a. OWL 1) is based on the DL $\mathcal{SHOIQ}$, while the more recent OWL 2 standardis based on $\mathcal{SROIQ}$ [OWL Working Group, 2009]. Unfortunately, the expressiveness of these DLs comes at the cost of high complexity of reasoning: most of the traditional DL reasoning tasks, like deciding knowledge base satisfiability or concept subsumption, are NExpTime-complete for $\mathcal{SHOIQ}$ [Tobies, 2000] and 2NExpTime complete for $\mathcal{SROIQ}$ [Kazakov, 2008]. If data complexity is considered (that is, if the complexity is measured in terms of the size of the assertional component while disregarding the size of the terminological component), then both logics are co-NP-hard. As more and more application areas require higher scalability, the study of fragments of expressive DLs with better computational properties has become an important area of research.

Horn fragments of DLs, which are obtained by restricting the syntax of a DL in such a way that disjunction is not expressible, were first considered in [Hustadt *et al.*, 2005] as expressive fragments with tractable data complexity (see also [Krötzsch *et al.*, 2007]). It was later identified that they can also exhibit lower combined complexity when it comes to query answering. Indeed, answering *conjunctive queries* (CQs), a kind of database-inspired queries that have become the standard for querying DLs, is in ExpTime for the Horn fragment of the prominent $\mathcal{SHIQ}$ [Eiter *et al.*, 2008], while it is already 2ExpTime-hard for quite restricted (non Horn) fragments of $\mathcal{SHIQ}$, like $\mathcal{ALCI}$ [Lutz, 2008] and $\mathcal{SH}$ [Eiter *et al.*, 2009]. It remained open whether this improvement in the combined complexity of query answering extends to more expressive logics, and in particular, to the Horn fragments of the DLs $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ that underlie the OWL standards. A positive answer is given in this paper.

The distinguishing feature of $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ is that they support *nominals* $\mathcal{O}$, which allow to restrict the cardinality of a concept, *inverses* $\mathcal{I}$, which allow to navigate roles in both directions, and *number restrictions* $\mathcal{Q}$, which allow to restrict the number of pairs of objects that may participate in a role. All of these features have been incorporated to satisfy practical modeling requirements posed by knowledge engineers. The interaction of these three constructors constitutes a severe obstacle to establishing the *forest model property* that is key to the design of algorithms. More often than not, this also causes an increased complexity of reasoning. Some problems that are well understood for most DLs remain open for $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$, most notably the decidability of query answering. It is only known that the problem is decidable (although no complexity bounds are known) for $\mathcal{ALCHOIQb}$ [Rudolph and Glimm, 2010], and already undecidable for $\mathcal{ALCOIF}_{reg}$ [Ortiz *et al.*, 2010a].

The Horn fragments of $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ remained unexplored until [Ortiz *et al.*, 2010b], where it was established that every satisfiable knowledge base in these languages has a forest-like model that can be represented in a relatively compact way. This and the lack of disjunction allowed the authors to show that traditional reasoning tasks are ExpTime-complete and 2ExpTime-complete in combined complexity for Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$, respectively, and are PTime-complete in data complexity.

*This work was partially supported by (†) the Austrian Science Fund (FWF) grant P20840, the Vienna Science and Technology Fund (WWTF) project ICT08-032, and (‡) the project ExpreSST funded by the Deutsche Forschungsgemeinschaft (DFG).

In this paper we develop an algorithm for answering *conjunctive 2-way regular path queries* (2CRPQs), a non-trivial generalization of plain CQs that allow for expressing complex navigation in models by means of regular expressions. They also generalize the (2-way) regular path queries used for querying web data and other forms of graph databases, and they constitute a large fragment of the SPARQL 1.1 query language for ontologies that is currently under standardization.[1] We show that a satisfiable KB in the Horn fragments of $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ has a *universal* model that can be used to answer all queries, and use a combination of query partitioning and tree automata techniques for query answering over a model representation similar to the one in [Ortiz *et al.*, 2010b]. Thereby we show that query answering is not more expensive than standard reasoning: ExpTime-complete for Horn-$\mathcal{SHOIQ}$ and 2ExpTime-complete for the more expressive Horn-$\mathcal{SROIQ}$ in combined complexity, but PTime-complete in data complexity for both. Notably, these are the first completeness results for query answering in DLs with inverses, nominals, and counting.

2 Preliminaries

We start by introducing the DLs and the query language considered in this paper.

The Horn Fragments of $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$

We recall the syntax of Horn-$\mathcal{SROIQ}$ and Horn-$\mathcal{SHOIQ}$, as defined in [Ortiz *et al.*, 2010b].[2]

We assume countably infinite sets $\mathsf{N_C} \supset \{\top, \bot\}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ of *concept names*, *role names*, and *individuals* respectively. If $r \in \mathsf{N_R}$, then r and r^- are *roles*, and their respective *inverses* are $\mathsf{inv}(r) = r^-$ and $\mathsf{inv}(r^-) = r$. We let $\overline{\mathsf{N_R}} = \mathsf{N_R} \cup \{r^- \mid r \in \mathsf{N_R}\}$. We assume a strict partial order $\prec$ on the roles such that $t \prec r$ iff $\mathsf{inv}(t) \prec r$. Expressions of the forms (i) to (xii) in Table 1 are called *axioms*; in particular, axioms of the forms (i) to (v) in Table 1 are *role inclusion axioms* (RIAs). The *non-simple roles* in a given set $\mathcal{R}$ of RIAs are inductively defined: (A) r is non-simple if there is a RIA $r_1 \circ \ldots \circ r_n \sqsubseteq r$ in $\mathcal{R}$ with $n > 1$, (B) r is non-simple if $\mathcal{R}$ contains a RIA of the form $s \sqsubseteq r$ with s non-simple, and (C) r^- is non-simple iff r is non-simple. Then a role occurring in $\mathcal{R}$ is *simple* if it is not non-simple.

A *terminology* $\mathcal{T}$ is a set of axioms. The *simple roles* in $\mathcal{T}$ are the simple roles in $\{\alpha \mid \alpha \text{ is a RIA in } \mathcal{T}\}$. A Horn-$\mathcal{SROIQ}$ knowledge base (KB) is a tuple $\mathcal{K}=(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a terminology and the *ABox* $\mathcal{A}$ is a non-empty set of *assertions* of the forms (xiii) and (xiv). The DL Horn-$\mathcal{SHOIQ}$ results from restricting Horn-$\mathcal{SROIQ}$ as follows: (A) allowing only RIAs of the forms (i), (ii') and (v); (B) disallowing *disjointness axioms* (vi); and (C) disallowing concepts of type $\exists t$.Self. Further restricting Horn-$\mathcal{SHOIQ}$ by allowing only expressions of types (i), (ii'), (vi)–(ix), (xi), (xiii) and (xiv) (marked in boldface in the table) yields the DL that we call Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$. Note that in such a KB all roles are simple. The semantics of KBs is defined in the usual, model-theoretic way via *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$.

Syntax		terminology		Syntax ABox	
(i)	$r^- \sqsubseteq r$	**(vii)**	$A \sqcap B \sqsubseteq C$	**(xiii)**	$a{:}A$
(ii)	$w \sqsubseteq r$	**(viii)**	$A \sqsubseteq \forall r.B$	**(xiv)**	$(a,b){:}t$
(ii')	$r_1 \sqsubseteq r$	**(ix)**	$A \sqsubseteq \exists r.B$		
(iii)	$r \circ w \sqsubseteq r$	(x)	$\exists r.A \sqsubseteq B$		
(iv)	$w \circ r \sqsubseteq r$	**(xi)**	$A \sqsubseteq \leqslant 1\, t.B$		
(v)	$r \circ r \sqsubseteq r$	(xii)	$A \sqsubseteq \geqslant m\, t.B$		
(vi)	$\mathsf{Disj}(t, t')$				

Table 1: Syntax of Horn-$\mathcal{SROIQ}$ in normal form. Here $m \geq 0$, $a, b \in \mathsf{N_I}$, r, r_i are roles, and t and t' are simple roles. Further, $w = r_1 \circ \ldots \circ r_n$ where $n \geq 1$ and, for each $1 \leq i \leq n$, holds $r_i \prec r$. *Concepts* A, B, and C are either concept names, *nominals* of the form $\{a\}$, or of the form $\exists t$.Self.

2-Way Conjunctive Regular Path Queries

The queries we consider use nondeterministic finite state automata (NFAs). Recall that an NFA is a tuple $\alpha = (\Sigma, S, \delta, s_0, F)$, where Σ is the *alphabet*, S is the set of *states*, $\delta \subseteq S \times \Sigma \times S$ is the *transition relation*, $s_0 \in S$ is the *initial state*, and $F \subseteq S$ is the set of *final states*. We use Σ^α, S^α, δ^α, s_0^α and F^α to denote the components of α.

Now we define *2-way conjunctive regular path queries* (*2CRPQ*, or simply *queries*), whose atoms use nondeterministic finite state automata (NFAs) to define regular languages over the alphabet $\overline{\mathsf{N_R}}$. Let V be a countably infinite set of variables. Then a 2CRPQ q is a set of atoms of the form $\alpha(x, y)$, where $x, y \in \mathsf{V}$ and α is an NFA with $\Sigma^\alpha = \overline{\mathsf{N_R}}$. We use $\mathsf{V}(q)$ to denote the set of variables occurring in q.

Let $\mathcal{I}$ be an interpretation and α an NFA with $\Sigma^\alpha = \overline{\mathsf{N_R}}$. An *$\alpha$-path* (from d_0 to d_n) in $\mathcal{I}$ is a sequence $d_0, \ldots, d_n$ of elements of $\Delta^{\mathcal{I}}$ such that for each $0 \leq i < n$ there is a state $s_i \in S^\alpha$ and a role $r_i \in \overline{\mathsf{N_R}}$ such that $(d_i, d_{i+1}) \in r_i^{\mathcal{I}}$, $s_0 = s_0^\alpha$, $s_{n-1} \in F^\alpha$ and $(s_i, r_i, s_{i+1}) \in \delta^\alpha$. We call *$d'$ an α-successor of d in $\mathcal{I}$* if there is an α-path from d to d'.

A *match* for a query q in an interpretation $\mathcal{I}$ is a mapping $\pi : \mathsf{V}(q) \to \Delta^{\mathcal{I}}$ such that $\pi(y)$ is an α-successor of $\pi(x)$ for each $\alpha(x, y) \in q$. We write $\mathcal{I} \models q$ if there is a match for q in $\mathcal{I}$. Given a knowledge base $\mathcal{K}$, we write $\mathcal{K} \models q$ if $\mathcal{I} \models q$ for every model $\mathcal{I}$ of $\mathcal{K}$.

Note that queries are *Boolean*, i.e., they do not allow for answer variables. They can simulate the standard (Boolean) conjunctive queries, that use concept and role names instead of NFAs in the atoms, and allow for individuals in the place of variables.[3] As usual, the decision problem associated to answering queries with answer variables can be reduced to the entailment of a Boolean query.

Simplifying Horn-$\mathcal{SROIQ}$ and Horn-$\mathcal{SHOIQ}$ KBs

Query answering in Horn-$\mathcal{SROIQ}$ and Horn-$\mathcal{SHOIQ}$ KBs can be reduced to query answering in the simpler Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ KBs. The following proposition is an easy consequence of the knowledge base transformation in [Kazakov, 2008]:

[1] `http://www.w3.org/TR/sparql11-query/`

[2] Due to space limitations, we focus on the normal form used in [Ortiz *et al.*, 2010b]. As usual, every KB can be polynomially rewritten into a normal one that is equivalent modulo the fresh concept names introduced by the translation, see [Krötzsch *et al.*, 2007].

[3] Individuals and concepts can be eliminated from the query in polynomial time using fresh query atoms and axioms in the KB.

Proposition 1. *For every Horn-$\mathcal{SROIQ}$ terminology $\mathcal{T}$ and query q there exists a Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ terminology $\mathcal{T}'$ and a query q' such that:*

(1) $(\mathcal{T}, \mathcal{A}) \models q$ iff $(\mathcal{T}', \mathcal{A}) \models q'$, for any ABox $\mathcal{A}$;

(2) $\mathcal{T}'$ and q' can be effectively constructed from $\mathcal{T}$ and q in single exponential time. If $\mathcal{T}$ is in Horn-$\mathcal{SHOIQ}$, then $\mathcal{T}'$ and q' can be constructed from $\mathcal{T}$ and q in polynomial time (assuming that numbers are encoded unary).

3 Query Answering

Assume in what follows a Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a query q. Our goal is to provide an algorithm and characterize the combined and data complexity of deciding $\mathcal{K} \models q$. By Proposition 1, such a procedure can be used for Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ as well. Without loss of generality we assume that $\mathcal{K}$ is satisfiable.[4]

3.1 Universal Model Property

Our method builds on the following crucial property: the KB $\mathcal{K}$ has a *universal* model, i.e. there is a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{K} \models q'$ iff $\mathcal{I} \models q'$, for any query q'. The universality of $\mathcal{I}$ is due to the guaranteed existence of a homomorphic embedding φ of $\mathcal{I}$ into any other model $\mathcal{I}'$ of $\mathcal{K}$. Intuitively, if $\mathcal{I}$ has a match π for q, then the composition $\varphi \circ \pi$ is a match for q in $\mathcal{I}'$.

By employing the results in [Ortiz *et al.*, 2010b] we can further show that $\mathcal{K}$ has a universal model $\mathcal{I}$ with a quite regular structure, which allows to finitely represent it. To this end, we introduce a representation of (possibly infinite) models of $\mathcal{K}$ via *model descriptions*. In the next section, we will see that model descriptions give rise to an optimal algorithm for query answering.

Intuitively, a model description contains a graph part and a successor function. The graph part comprises a finite number of domain elements $\mathbf{G}$ that may be arbitrarily interrelated by roles, while the successor function tells us how this finite structure can be expanded into a potentially infinite forest-shaped structure.

Definition 1 (Model description). A *concept type* τ (for $\mathcal{T}$) is any set of concepts occurring in the axioms of $\mathcal{T}$ and that does not contain $\bot$ and does not contain a pair of concepts of the form $\{\leqslant 1\, t.B, \geqslant m\, t.B\}$ with $m > 1$. A *role-type* ρ (for $\mathcal{T}$) is a set of roles occurring in $\mathcal{T}$. The set of all concept types of $\mathcal{T}$ is denoted by $\mathbf{T}_\mathcal{T}$, and the set of all role types by $\mathbf{R}_\mathcal{T}$.

A *model description for* $\mathcal{K}$ is a tuple $\mathcal{M} = (\mathbf{G}, \theta, f, succs)$ where:

(a) $\mathbf{G}$ is a set of objects called *graph nodes*;

(b) θ is a function that maps each $d \in \mathbf{G}$ to a concept type $\theta(d) \in \mathbf{T}_\mathcal{T}$ and each pair $(d, d') \in \mathbf{G} \times \mathbf{G}$ to a role type $\theta(d, d') \in \mathbf{R}_\mathcal{T}$ in such a way that $\theta(d, d') = \{\mathsf{inv}(r) \mid r \in \theta(d, d')\}$ for each $d, d' \in \mathbf{G}$;

(c) f is a function $f : \mathsf{N_I}(\mathcal{A}) \to \mathbf{G}$, where $\mathsf{N_I}(\mathcal{A})$ denotes the set of individuals occurring in $\mathcal{A}$;

(d) $succs : \mathbf{G} \cup (\mathbf{T}_\mathcal{T} \times \mathbf{R}_\mathcal{T} \times \mathbf{T}_\mathcal{T}) \to 2^{\mathbf{R}_\mathcal{T} \times (\mathbf{T}_\mathcal{T} \cup \mathbf{G})}$ is a function such that $succs(d) \in 2^{\mathbf{R}_\mathcal{T} \times \mathbf{T}_\mathcal{T}}$ for each $d \in \mathbf{G}$.

We use $\mathsf{receive}(\mathcal{M})$ to denote the nodes in $\mathbf{G}$ that occur in the range of $succs$.

Intuitively, every model description represents some interpretation $\mathcal{I}_\mathcal{M}$. Its domain contains the graph nodes $\mathbf{G}$, and each graph node is the root of a possibly infinite tree-shaped structure in which some nodes may be related to graph nodes (the latter 'receiving' nodes are represented by $\mathsf{receive}(\mathcal{M})$). Each $d \in \mathbf{G}$ is associated to a concept type $\theta(d)$ that contains precisely the concepts it satisfies, and each pair d, d' to a role type $\theta(d, d')$ containing precisely the roles that relate them. The function f tells which individual names are interpreted as which graph nodes, while $succs$ describes how tree-shaped structures can be generated starting from graph nodes. To understand this, recall that $succs$ is defined for elements in $d \in \mathbf{G}$ and triplets $(\tau_1, \rho, \tau_2) \in \mathbf{T}_\mathcal{T} \times \mathbf{R}_\mathcal{T} \times \mathbf{T}_\mathcal{T}$. This means that $succs$ gives a specific successor configuration for the graph nodes, while for the remaining nodes the children are given based on a *concepts-roles-concepts* pattern. In particular, the children of a tree node e are determined based on the type of e, the type of its parent and the roles that connect them. The children are given as a set of tuples $(\rho, \tau) \in \mathbf{R}_\mathcal{T} \times \mathbf{T}_\mathcal{T}$ and $(\rho, d) \in \mathbf{R}_\mathcal{T} \times \mathbf{G}$. Intuitively, (ρ, τ) stands for a ρ-child satisfying τ, while (ρ, d) means a ρ-link to the graph node d. We build $\mathcal{I}_\mathcal{M}$ formally as follows:

Definition 2. Let $\mathcal{M} = (\mathbf{G}, \theta, f, succs)$ be a model description for $\mathcal{K}$. We say that $\mathcal{M}$ *describes the interpretation* $\mathcal{I}_\mathcal{M}$ defined as follows:

The domain $\Delta^{\mathcal{I}_\mathcal{M}}$ is the smallest set of words of the form $e = d\rho_1\tau_2 \ldots \rho_{n-1}\tau_n \in \mathbf{G} \times (\mathbf{T}_\mathcal{T} \times \mathbf{R}_\mathcal{T})^*$ such that:

- $\mathbf{G} \subseteq \Delta^{\mathcal{I}_\mathcal{M}}$,
- if $d \in \mathbf{G}$ and $(\rho', \tau') \in succs(d)$, then $d\rho'\tau' \in \Delta^\mathcal{I}$,
- if $e\rho\tau \in \Delta^{\mathcal{I}_\mathcal{M}}$ and $(\rho', \tau') \in succs(\mathsf{last}(e), \rho, \tau)$, then $e\rho\tau\rho'\tau' \in \Delta^\mathcal{I}$,

where $\mathsf{last}(e) = \theta(e)$ if $e \in \mathbf{G}$, and $\mathsf{last}(e\rho\tau) = \tau$ otherwise.

Nodes in $\Delta^{\mathcal{I}_\mathcal{M}} \setminus \mathbf{G}$ are called *tree nodes*, and we say that a node $e\rho\tau \in \Delta^{\mathcal{I}_\mathcal{M}}$ is a *child* of the node e.

The interpretation function $\cdot^{\mathcal{I}_\mathcal{M}}$ is defined next:

- for each $a \in \mathsf{N_I}(\mathcal{A})$, $a^{\mathcal{I}_\mathcal{M}} = f(a)$;
- for each $A \in \mathsf{N_C}$, $A^{\mathcal{I}_\mathcal{M}} = \{e \mid A \in \mathsf{last}(e)\}$;
- for each $r \in \mathsf{N_R}$, $r^{\mathcal{I}_\mathcal{M}}$ is the set of pairs that contains:
  - (d, d') if $d, d' \in \mathbf{G}$ and $r \in \theta(d, d')$
  - $(e, e\rho\tau)$ whenever $e, e\rho\tau \in \Delta^\mathcal{I}$ and $r \in \rho$,
  - $(e\rho\tau, e)$ whenever $e, e\rho\tau \in \Delta^\mathcal{I}$ and $\mathsf{inv}(r) \in \rho$,
  - (e, e) whenever $\exists r.\mathsf{Self} \in \mathsf{last}(e)$,
  - $(e\rho\tau, d)$ if $d \in \mathbf{G}$ and $(\rho', d) \in succs(\mathsf{last}(e), \rho, \tau)$ for a ρ' with $r \in \rho'$,
  - $(d, e\rho\tau)$ if $d \in \mathbf{G}$ and $(\rho', d) \in succs(\mathsf{last}(e), \rho, \tau)$ for a ρ' with $\mathsf{inv}(r) \in \rho$.

Importantly, there is a model description that represents the desired universal model, and it can be constructed in time that is single exponential in the size $\|\mathcal{K}\|$ and only polynomial in $\|\mathcal{A}\|$ (for fixed $\mathcal{T}$):

[4] If $\mathcal{K}$ is unsatisfiable, then $\mathcal{K} \models q$ for every q. Satisfiability of $\mathcal{K}$ within the necessary time bounds can be checked using the algorithm in [Ortiz *et al.*, 2010b].

Theorem 2. *For the KB $\mathcal{K}$ there is a model description $\mathcal{M} = (\mathbf{G}, \theta, f, succs)$ such that:*

(1) $\mathcal{I}_\mathcal{M} \models \mathcal{K}$;

(2) $\mathcal{K} \models q'$ iff $\mathcal{I}_\mathcal{M} \models q'$, for every query q';

(3) $\mathcal{M}$ can be build in time single exponential in $\|\mathcal{K}\|$ (and is thus of size at most exponential in $\|\mathcal{K}\|$). If $\mathcal{T}$ is fixed, then $\mathcal{M}$ can be obtained in polynomial time in $\|\mathcal{A}\|$.

(4) If $\mathcal{T}$ is fixed, then $|\mathsf{receive}(\mathcal{M})|$ is bounded by a constant.

Sketch. In [Ortiz *et al.*, 2010b] it was shown that a model of a given Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ KB $\mathcal{K}$ can be obtained from the minimal model I of a Datalog program obtained from $\mathcal{K}$. It can be shown in a rather straightforward way that the resulting model is universal. The claim follows from those results, since the I is very closely related to the model descriptions used in this paper. To make this paper as self-contained as possible, we sketch here a direct proof of existence and universality of the mentioned model representation.

Let $\mathfrak{M}$ be the set of all models of $\mathcal{K}$. We then define $\mathcal{M}$ as follows (using $\tau^\mathcal{I}$ as a shortcut for $\bigcap_{A\in\tau} A^\mathcal{I}$):

(a) To define $\mathbf{G}$, we first let $\hat{\mathbf{T}}$ denote the set of all $\tau \in \mathbf{T}_\mathcal{T}$ for which (i) every $\mathcal{I} \in \mathfrak{M}$ satisfies $|\tau^\mathcal{I}| = 1$ and (ii) there is no $\tau' \supset \tau$ with the same property (i.e. τ is inclusion-maximal).

We define the equivalence relation $\approx$ on $\mathsf{N_I}(\mathcal{A})$ by letting $a \approx b$ iff $a^\mathcal{I} = b^\mathcal{I}$ for all $\mathcal{I} \in \mathfrak{M}$. Now we let $\mathbf{G}$ contain all pairs (c, τ') for which either (i) c is a $\approx$-equivalence class and, for any $b \in c$, $\tau' = \bigcap_{\mathcal{I}\in\mathfrak{M}}\{A \mid b^\mathcal{I} \in A^\mathcal{I}\}$ (we call these the *named graph nodes*) or (ii) $c = \emptyset$, $\tau' \in \hat{\mathbf{T}}$ and there is no $b \in \mathsf{N_I}(\mathcal{A})$ such that $b^\mathcal{I} \in \tau^\mathcal{I}$ for all $\mathcal{I} \in \mathfrak{M}$ (we call these the *anonymous graph nodes*).

(b) For each $(c, \tau') \in \mathbf{G}$, we define θ by letting $\theta((c, \tau')) := \tau'$ and letting $\theta((c_1, \tau_1), (c_2, \tau_2))$ contain all roles r for which every $\mathcal{I} \in \mathfrak{M}$ satisfies that $(e_1, e_2) \in r^\mathcal{I}$ where $e_i \in \Delta^\mathcal{I}$ is the domain element with $e_i \in \tau_i^\mathcal{I}$ and $e_i = d^\mathcal{I}$ for all $d \in c_i$.

(c) f is such that every b is mapped to the unique $([b]_\approx, \tau')$.

(d) We define *succs* as follows: For $d \in \mathbf{G}$, we let $succs(d)$ contain all (ρ, τ') where (i) for every $\mathcal{I} \in \mathfrak{M}$ and $e \in \tau^\mathcal{I}$ there exists some $e' \in \tau'^\mathcal{I}$ with $(e, e') \in r^\mathcal{I}$ for all $r \in \rho$ and (ρ, τ') is inclusion-maximal w.r.t. that property; (ii) there is no $d' \in \mathbf{G}$ with $\rho \subseteq \theta(d, d')$ and $\tau' \subseteq \theta(d')$; (iii) $\{\exists r.\mathsf{Self} \mid r \in \rho\} \cup \tau' \not\subseteq \tau$. For $\tau, \tau' \in \mathbf{T}_\mathcal{T}$ and $\rho \in \mathbf{R}_\mathcal{T}$ we let $succs(\tau, \rho, \tau')$ contain

- all (ρ', τ'') where (i) for every $\mathcal{I} \in \mathfrak{M}$ and $e \in \tau'^\mathcal{I}$ there exists some $e' \in \tau''^\mathcal{I}$ with $(e, e') \in r^\mathcal{I}$ for all $r \in \rho'$ and (ρ', τ'') is inclusion-maximal w.r.t. that property; (ii) it is not the case that both $\rho' \subseteq \{\mathsf{inv}(r) \mid r \in \rho\}$ and $\tau'' \subseteq \tau$; (iii) there is no $(c, \tau''') \in \mathbf{G}$ with $\tau'' \subseteq \tau'''$; (iv) $\{\exists r.\mathsf{Self} \mid r \in \rho\} \cup \tau' \not\subseteq \tau$;
- all $(\rho', (c, \tau''))$ with $(c, \tau'') \in \mathbf{G}$ where for every $\mathcal{I} \in \mathfrak{M}$, $e \in \tau'^\mathcal{I}$, $e' \in \tau''^\mathcal{I}$ and $r \in \rho'$ holds $(e, e') \in r^\mathcal{I}$ and $\tau'' \in \hat{\mathbf{T}}$.

By inspecting each type of axiom that may occur in $\mathcal{K}$ (see Table 1), one can show that the interpretation $\mathcal{I}_\mathcal{M}$ defined in Definition 2 is a model of $\mathcal{K}$ as claimed in 1.

To prove that $\mathcal{I}_\mathcal{M}$ is universal (Claim 2), one can show that there is a homomorphism φ from $\mathcal{I}_\mathcal{M}$ to every model $\mathcal{I}$ of $\mathcal{K}$, which can be constructed inductively on the length of the elements of $\Delta^{\mathcal{I}_\mathcal{M}}$ as follows:

- We start from $\mathbf{G}$. For each named graph node $([a]_\approx, \tau')$, we let $\varphi(([a]_\approx, \tau')) := a^\mathcal{I}$. For each anonymous graph node $(\emptyset, \tau')$, we let $\varphi((\emptyset, \tau')) = w$ for the unique w with $\{w\} = \tau^\mathcal{I}$.
- Let each $e\rho\tau \in \Delta^{\mathcal{I}_\mathcal{M}}$ for which $\varphi(e)$ has been defined, we set $\varphi(e\rho\tau) = w$ for a w with $w \in \bigcap_{A\in\tau} A^\mathcal{I}$ and $(\varphi(e), w) \in \bigcap_{r\in\rho} r^\mathcal{I}$; such a w exists in every model of $\mathcal{K}$ by the construction of $\mathcal{M}$.

The existence of the homomorphism φ into every model $\mathcal{I}$ ensures that whenever there is a match π for a 2CRPQ q in $\mathcal{I}_\mathcal{M}$, then $\pi \circ \varphi$ is a match for q in $\mathcal{I}$. This shows that $\mathcal{I}_\mathcal{M}$ is universal, i.e. $\mathcal{K} \models q$ iff $\mathcal{I}_\mathcal{M} \models q$.

To see that $\mathcal{M}$ satisfies Claim 4, note that the number of distinct $(c, \tau) \in \mathbf{G}$ satisfying $\tau'' \in \hat{\mathbf{T}}$ for every $\mathcal{I} \in \mathfrak{M}$ (let us call these particular elements $\mathbf{G}^\mathsf{s}$) is exponentially bounded by $\mathcal{T}$, since there can be only one such element for each τ. Yet, by definition only elements of $\mathbf{G}^\mathsf{s}$ occur in the range of *succs* and hence in $\mathsf{receive}(\mathcal{M})$.

To prove Claim 3 note that the exponential bound on the size of $\mathcal{M}$ follows from the fact that there are at most exponentially many different types. Likewise, this gives rise to an exponential time algorithm since all (overall single exponentially many) decisions to define $\mathcal{M}$ requiring inspections of the set $\mathfrak{M}$ of all models can be cast into Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ standard reasoning tasks which are known to be ExpTime-complete.

For the polynomial size bound of $\mathcal{M}$ w.r.t. $|\mathcal{A}|$ note that we have by definition $|\mathbf{G} \setminus \mathsf{receive}(\mathcal{M})| \leq |\mathsf{N_I}(\mathcal{A})|$, consequently the sizes of the $\mathsf{receive}(\mathcal{M})$-related parts of *succs* and θ are at most linearly and quadratically bounded by $|\mathsf{N_I}(\mathcal{A})|$, respectively. Further, for a fixed $\mathcal{T}$, $\mathsf{receive}(\mathcal{M})$ and the respective parts of *succs* and θ are constant, hence $|\mathcal{M}|$ is of polynomial size w.r.t. $|\mathcal{A}|$ as claimed. The polynomial bound on the time needed to establish $\mathcal{M}$ is a consequence of the known time-polynomial data complexity of Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ as the conditions for the definition of $\mathbf{G}^\mathsf{s}$ and the respective parts of *succs* and θ can be obtained by a polynomial number of Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ satisfiability checks on $\mathcal{K}$. □

3.2 Algorithm and Complexity

By Theorem 2, deciding $\mathcal{K} \models q$ is equivalent to checking $\mathcal{I}_\mathcal{M} \models q$, where $\mathcal{M}$ is a model description for a universal model of $\mathcal{K}$. We now assume such an $\mathcal{M} = (\mathbf{G}, \theta, f, succs)$, and provide a method to check $\mathcal{I}_\mathcal{M} \models q$.

First we define the tree-shaped parts of $\mathcal{I}_\mathcal{M}$.

Definition 3. For each $d \in \mathbf{G}$ the *d-tree* of $\mathcal{I}_\mathcal{M}$, denoted $\mathcal{I}_{\mathcal{M},d}$, is obtained from $\mathcal{I}_\mathcal{M}$ in two steps by (1) restricting $\mathcal{I}_\mathcal{M}$ to the domain $\Delta^{\mathcal{I}_{\mathcal{M},d}}$ that contains (a) each $e \in \Delta^{\mathcal{I}_\mathcal{M}}$ of the form $d\rho\tau \dots \rho'\tau'$, and (b) each $d' \in \mathbf{G}$ such that $(e, d') \in r^{\mathcal{I}_\mathcal{M}}$ for some $e = d\rho\tau \dots \rho'\tau'$ as above and some $r \in \overline{\mathsf{N_R}}$; and (2) setting $\{r \mid (d, d') \in r^\mathcal{I}\} = \emptyset$ for all $d, d' \in \Delta^{\mathcal{I}_{\mathcal{M},d}} \cap \mathbf{G}$.

To find a match π for q in $\mathcal{I}_\mathcal{M}$, we first consider a function δ that 'guesses' for each variable x of q whether $\pi(x)$ is a graph node $d = \delta(x)$, or a tree node inside a d-tree (indicated by $\delta(x) = d_\downarrow$). Based on δ, we decompose q into several queries, additionally guessing for each $\alpha(x, y)$ of q which of three kinds of paths exists between $\pi(x)$ and $\pi(y)$: a path of

type p that starts and ends in a graph node, a path t that is entirely inside a d-tree and does not pass any other graph node, or a path that starts or ends inside a tree, but passes at least one other graph node on the way.

Definition 4 (Query Partition). Let $\mathbf{S} = \bigcup_{\alpha(x,y)\in q} S^\alpha$, and let $\mathbf{G}' = \mathbf{G} \cup \{d_\downarrow \mid d \in \mathbf{G}\}$. A *partition* Π *of* q is a pair (δ, Γ) where $\delta : \mathsf{Vars}(q) \to \mathbf{G}'$ and $\Gamma : q \to (\mathbf{G} \times \mathbf{S})^2 \cup \{\mathsf{p}, \mathsf{t}\}$ such that one of the following holds for each $\alpha(x, y) \in q$:

- $\delta(x), \delta(y) \in \mathbf{G}$ and $\Gamma(\alpha(x, y)) = \mathsf{p}$, or
- there is some $d \in \mathbf{G}$ such that $\delta(x), \delta(y) \in \{d, d_\downarrow\}$, and $\Gamma(\alpha(x, y)) = \mathsf{t}$,
- $\{\delta(x), \delta(y)\} \not\subseteq \mathbf{G}$ and $\Gamma(\alpha(x, y)) = \langle (d_1, s_1), (d_2, s_2)\rangle \in (\mathbf{G} \times \mathbf{S})^2$ with $s_1, s_2 \in Q_\alpha$, and $\delta(y) \in \{d_2, d_{2\downarrow}\}$.

For the parts of the paths that start or end inside the trees, we build queries that are evaluated over the d-trees. We consider atoms $\alpha(t, t')$ with $t, t' \in \mathbf{G} \cup \mathsf{V}$. A match π for such a query q in a part of $\mathcal{I}_\mathcal{M}$ is defined analogously as in Section 2, but it must additionally map each $d \in \mathbf{G}$ occurring in q to itself. For an NFA α, $s \in S^\alpha$ and $S \subseteq S^\alpha$, we define $\alpha_{s,S} = (\Sigma^\alpha, S^\alpha, \delta^\alpha, s, S)$, that is, $\alpha_{s,S}$ is obtained by making s the initial state and S the set of final states of α.

Definition 5. For each $d \in \mathbf{G}$, the *d-query* of Π is the set $tq(\Pi, d)$ containing

- each $\alpha(f(x), f(y))$ such that $\alpha(x, y) \in q$, $\delta(x), \delta(y) \in \{d, d_\downarrow\}$, and $\Gamma(\alpha(x, y)) = \mathsf{t}$;
- each $\alpha_{s_0^\alpha,\{s_1\}}(f(x), d_1)$ such that $\alpha(x, y) \in q$, $\delta(x) \in \{d, d_\downarrow\}$, and $\Gamma(\alpha(x, y)) = \langle (s_1, d_1), (s_2, d_2)\rangle$; and
- each $\alpha_{s_2,F^\alpha}(d_2, f(y))$ such that $\alpha(x, y) \in q$, $\delta(y) \in \{d_2, d_{2\downarrow}\}$, and $\Gamma(\alpha(x, y)) = \langle (s_1, d_1), (s_2, d_2)\rangle$,

where, for a variable x, $f(x) = \delta(x)$ if $\delta(x) \in \mathbf{G}$, and $f(x) = x$ otherwise.

For the remaining parts of the paths, we rely on a relation $\mathsf{paths}(\mathcal{M})$ that contains all the relevant paths between graph nodes. We will show later how this relation can be effectively computed.

Definition 6. We define $\mathsf{paths}(\mathcal{M}) = \{\langle (s, d)(s', d')\rangle \in (\mathbf{G} \times \mathbf{S})^2 \mid d' \text{ is an } \alpha_{s,\{s'\}}\text{-successor of } d \text{ in } \mathcal{I}_\mathcal{M}\}$. Let $\Pi = (\delta, \Gamma)$ be a partition of q. Then we say that $\mathsf{paths}(\mathcal{M})$ *satisfies the graph paths* of Π if the following hold:

- for each $\alpha(x, y) \in q$ such that $\{\delta(x), \delta(y)\} \subseteq \mathbf{G}$ and $\Gamma(\alpha(x, y)) = \mathsf{p}$, there is some $s_f \in F^\alpha$ such that $\langle (\delta(x), s_0^\alpha), (\delta(y), s_f)\rangle \in \mathsf{paths}(\mathcal{M})$, and
- for each $\alpha(x, y) \in q$ such that $\Gamma(\alpha(x, y)) = \langle (d, s)(d's')\rangle$, $\langle (d, s)(d's')\rangle \in \mathsf{paths}(\mathcal{M})$.

We write $\mathcal{M} \models \Pi$ if $\mathsf{paths}(\mathcal{M})$ satisfies the graph paths of Π, and $\mathcal{I}_{\mathcal{M},d} \models tq(\Pi, d)$ for every $d \in \mathbf{G}$.

Every match for q in $\mathcal{I}_\mathcal{M}$ can be used to define a partition such that $\mathcal{M} \models \Pi$. Conversely, if $\mathcal{M} \models \Pi$ then there is a match for the full q in $\mathcal{I}_\mathcal{M}$. Hence we have:

Theorem 3. *$\mathcal{I}_\mathcal{M} \models q$ iff there exists query partition Π of q such that $\mathcal{M} \models \Pi$.*

Consequently, $\mathcal{I}_\mathcal{M} \models q$ (and $\mathcal{K} \models q$), reduces to finding a query partition Π such that $\mathcal{M} \models \Pi$. We first note that the number of distinct partitions Π of q is $O\big(|\mathbf{G}|^{|\mathsf{Vars}(q)|} \cdot (|\mathbf{G}| \cdot |\mathbf{S}|)^{2|q|}\big)$. Keeping in mind Theorem 2, the number of distinct partitions Π to be traversed is exponential in $\|\mathcal{K}\| + \|q\|$, but polynomial in $\|\mathcal{A}\|$ if $\mathcal{T}$ and q are fixed.

It remains to see how to decide $\mathcal{M} \models \Pi$. To this aim, we first show how $\mathcal{I}_{\mathcal{M},d} \models tq(\Pi, d)$ can be checked for a given partition Π and a $d \in \mathbf{G}$. We employ *automata over infinite trees*, which are a generalization of finite state automata over finite words [Vardi, 1998]. Recall that an *infinite k-ary tree over an alphabet* Σ is a pair $(T, \mathcal{L})$, where (a) T is the set of all finite words over $\{1, \ldots, k\}$; (b) $\mathcal{L} : T \to \Sigma$ is the labeling function that assigns to each node in T a symbol from Σ. Each tree automaton A is defined for some alphabet Σ and *accepts* some set $L(\mathrm{A})$ of infinite trees over Σ. The basic reasoning problem is to decide given an automaton A whether A accepts at least one tree, i.e. that $L(\mathrm{A}) \neq \emptyset$. Various operations on automata can be performed, e.g. for a pair of automata A_1, A_2, one can build the intersection automaton $\mathrm{A}_1 \cap \mathrm{A}_2$ such that $L(\mathrm{A}_1 \cap \mathrm{A}_2) = L(\mathrm{A}_1) \cap L(\mathrm{A}_2)$.

Let $q' = tq(\Pi, d)$. It is not too difficult to build a tree automaton A^* such that (a) $\mathcal{I}_{\mathcal{M},d} \models q'$ iff $L(\mathrm{A}^*) \neq \emptyset$, i.e. to reduce our problem to testing nonemptiness of the language of a tree automaton, (b) $L(\mathrm{A}^*) \neq \emptyset$ can be checked in time exponential in $\|\mathcal{K}\|+\|q\|$, and in polynomial time in $\|\mathcal{A}\|$ if $\mathcal{T}$ and q are fixed. Naturally, such a reduction involves an encoding of (tree-shaped) interpretations as labeled trees. The automaton A^* is defined as the intersection automaton $\mathrm{A}^* = \mathrm{A}_{\mathcal{M},d} \cap \mathrm{A}_{q'}$, where $\mathrm{A}_{\mathcal{M},d}$ accepts exactly one labeled tree that encodes the interpretation $\mathcal{I}_{\mathcal{M},d}$, and $\mathrm{A}_{q'}$ accepts labeled trees that encode interpretations with a match for q'. We next describe the languages $L(\mathrm{A}_{\mathcal{M},d})$ and $L(\mathrm{A}_{q'})$, and discuss the relevant properties of $\mathrm{A}_{\mathcal{M},d}$ and $\mathrm{A}_{q'}$, but omit their explicit construction.

The language $L(\mathrm{A}_{\mathcal{M},d})$ consists of exactly one labeled tree $(T, \mathcal{L})$ which represents $\mathcal{I}_{\mathcal{M},d}$. The alphabet is $\Sigma^\mathcal{M} = \Sigma_1^\mathcal{M} \cup \Sigma_2^\mathcal{M} \cup \{\mathsf{nil}\}$, where $\Sigma_1^\mathcal{M} = \mathbf{R}_\mathcal{T} \times \mathbf{T}_\mathcal{T}$ and $\Sigma_2^\mathcal{M} = \mathbf{R}_\mathcal{T} \times \mathsf{receive}(\mathcal{M})$. Intuitively, a node $w \in T$ labeled with $(\rho, \tau) \in \Sigma_1^\mathcal{M}$ corresponds to a tree node e_w in $\mathcal{I}_{\mathcal{M},d}$ with type τ and which is connected to its parent via the roles in ρ. On the other hand, a node $w{\cdot}c \in T$ labeled with $(\rho, d) \in \Sigma_2^\mathcal{M}$ reflects the roles connecting the tree node e_w in $\mathcal{I}_{\mathcal{M},d}$ to the graph node d. Technically, all nodes of T must have the same number of children, and thus we may need to use the nil label to nodes that do not correspond to elements in $\mathcal{I}_{\mathcal{M},d}$. $\mathrm{A}_{\mathcal{M},d}$ can be built as a deterministic 1-way tree automaton with state set $Q = \mathbf{T}_\mathcal{T} \times \mathbf{R}_\mathcal{T} \times \mathbf{T}_\mathcal{T} \cup \{d\}$ whose transition function δ is a notational reformulation of *succs*.

The language $L(\mathrm{A}_{q'})$ consists of trees over $\Sigma^\mathcal{M}$ that encode tree-shaped interpretations with a match for q'. To this end, let X be the set of variables in q'. Then the automaton $\mathrm{A}_{q'}$ can be built in two stages. First, we can build automaton A' that accepts trees $(T, \mathcal{L})$ over an extended alphabet $\Sigma' = 2^X \times \Sigma^\mathcal{M}$ with the following properties: (a) for each $x \in X$ there is exactly one node $w \in T$ with $\mathcal{L}(w) = (V, N)$ and $x \in V$, i.e. each variable $x \in X$ occurs in exactly one node of $(T, \mathcal{L})$ and thus $(T, \mathcal{L})$ encodes an interpretation together a variable assign-

ment π; (b) the variable assignment π witness a match for q' in the encoded interpretation. By exploiting the bound (4) in Theorem 2, we can implement A' as an alternating 2-way automaton whose state set is linear in $\|\mathcal{K}\| + \|q\|$, but which is constant if $\mathcal{T}$ and q are fixed. The automaton $\mathrm{A}_{q'}$ is then obtained from A' by transforming A' into a nondeterministic 1-way automaton [Vardi, 1998] and *projecting* away the first component in the labels of nodes, i.e. $L(\mathrm{A}_{q'})$ is the set of trees $(T, \mathcal{L})$ over $\Sigma^{\mathcal{M}}$ which can be endowed with a variable assignment such that resulting tree is accepted by A'.

Since the size of $q' = tq(\Pi, d)$ is polynomial in q, the number of states in the automata $\mathrm{A}_{\mathcal{M},d}$, $\mathrm{A}_{q'}$ and A^* is exponential in $\|\mathcal{K}\| + \|q\|$, however it is bounded by a constant if $\mathcal{T}$ and q are fixed. Using well-known algorithms we obtain that testing non-emptiness of A^* is feasible in time exponential in $\|\mathcal{K}\| + \|q\|$, and polynomial in $\|\mathcal{A}\|$ if $\mathcal{T}$ and q are fixed.

To obtain an algorithm for $\mathcal{M} \models \Pi$, it is only left to show how to decide whether $\mathsf{paths}(\mathcal{M})$ satisfies the graph paths of a given Π. We start by providing an effective construction of $\mathsf{paths}(\mathcal{M})$, which can be obtained as the smallest set of pairs that satisfies the following:

(a) For each $d, d' \in \mathbf{G}$, $\alpha(x, y) \in q$, $s, s' \in S^\alpha$, and $r \in \overline{\mathsf{N_R}}$, if $r \in \theta(d, d')$ and $(s, r, s') \in \delta^\alpha$ then $\langle (d, s), (d', s') \rangle \in \mathsf{paths}(\mathcal{M})$.

(b) For each $d, d' \in \mathbf{G}$, $\alpha(x, y) \in q$, and $s, s' \in S^\alpha$, if $\mathcal{I}_{\mathcal{M},d} \models \alpha_{s,\{s'\}}(d, d')$, then $\langle (d, s), (d', s') \rangle \in \mathsf{paths}(\mathcal{M})$.

(c) For $d, d', d'' \in \mathbf{G}$, $\alpha(x, y) \in q$ and $s, s', s'' \in S^\alpha$, if $\langle (s, d)(s', d') \rangle \in \mathsf{paths}(\mathcal{M})$ and $\langle (s', d')(s'', d'') \rangle \in \mathsf{paths}(\mathcal{M})$, then $\langle (s, d)(s'', d'') \rangle \in \mathsf{paths}(\mathcal{M})$.

There are at most $(|\mathbf{G}| \cdot |\mathbf{S}|)^2$ cases for which we have to test the conditions (a) and (b). Each test of the first kind can be done with a simple inspection of $\mathcal{M}$ and the NFAs in q, and for each test of the second type we can use the same automata-based decision procedure that we use for the d-queries above. Hence each test can be done in time that is exponential in $\|\mathcal{K}\| + \|q\|$, and in polynomial time in $\|\mathcal{A}\|$ if $\mathcal{T}$ and q are fixed. Finally, we close the table under condition (c) by a simple iterative procedure that reaches a fixed-point after at most $(|\mathbf{G}| \cdot |\mathbf{S}|)^{|q|}$ steps. Hence $\mathsf{paths}(\mathcal{M})$ can be effectively constructed in time that is exponential in $\|\mathcal{K}\| + \|q\|$, and in polynomial time in $\|\mathcal{A}\|$ if $\mathcal{T}$ and q are fixed.

Simple look-ups on this table allow us to decide whether $\mathsf{paths}(\mathcal{M})$ satisfies the graph paths of Π for any given Π. Hence, putting the above pieces together, we get:

Lemma 4. *Entailment of 2CRPQs in Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ is in* ExpTime *w.r.t. combined complexity and in* PTime *w.r.t. data complexity.*

We can now show the main complexity result of this paper:

Theorem 5. *Entailment of 2CRPQs in Horn-$\mathcal{SHOIQ}$ is* ExpTime*-complete, while the problem in Horn-$\mathcal{SROIQ}$ is* 2ExpTime*-complete with respect to combined complexity. The data complexity of both problems is* PTime*-complete.*

Proof. (Sketch) Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a Horn-$\mathcal{SHOIQ}$ (resp. Horn-$\mathcal{SROIQ}$) KB and q a query. Then the upper bounds follow from Lemma 4 and Proposition 1, which states that we can obtain a Horn-$\mathcal{ALCHOIQ}^{\mathsf{Disj}}_{\mathsf{Self}}$ KB $\mathcal{K}' = (\mathcal{T}', \mathcal{A})$ and a query q' such that: (a) $\mathcal{K} \models q$ iff $\mathcal{K}' \models q'$; (b) $\|\mathcal{K}'\| + \|q'\|$ is polynomial (resp., exponential) in $\|\mathcal{K}\| + \|q\|$; (c) if $\|\mathcal{T}\| + \|q\|$ is fixed, then $\|\mathcal{T}'\| + \|q'\|$ is fixed.

The lower bounds follow from the complexity of standard reasoning: the ExpTime and the PTime lower bounds can be found in [Hustadt *et al.*, 2005], while the 2ExpTime lower bound follows from [Ortiz *et al.*, 2010b]. □

4 Discussion and Conclusion

Our results show that answering complex queries like 2CRPQs in Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ is not just decidable but in fact even not any harder than standard reasoning, both in combined and data complexity. That CQ entailment checking is significantly easier in the Horn fragments than in the full DLs is arguably due to the existence of a universal model which can be conveniently represented and harnessed for efficient algorithms. We believe that the techniques to establish the result and the insights thus obtained will pave the way toward more efficient querying algorithms for the Horn restriction of OWL and might prove useful to tackle the problem of CQ entailment for full $\mathcal{SROIQ}$ and $\mathcal{SHOIQ}$.

References

[Eiter *et al.*, 2008] T. Eiter, G. Gottlob, M. Ortiz, and M. Šimkus. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proc. JELIA'08*, pages 166–179, 2008.

[Eiter *et al.*, 2009] T. Eiter, C. Lutz, M. Ortiz, and M. Šimkus. Query answering in description logics with transitive roles. In *Proc. IJCAI'09*, pages 759–764, 2009.

[Hustadt *et al.*, 2005] U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. IJCAI'05*, pages 466–471, 2005.

[Kazakov, 2008] Y. Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Proc. KR'08*, pages 274–284, 2008.

[Krötzsch *et al.*, 2007] M. Krötzsch, S. Rudolph, and P. Hitzler. Complexity boundaries for Horn description logics. In *Proc. AAAI'07*, pages 452–457, 2007.

[Lutz, 2008] Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In *Proc. IJCAR'08*, pages 179–193, 2008.

[Ortiz *et al.*, 2010a] M. Ortiz, S. Rudolph, and M. Šimkus. Query answering is undecidable in dls with regular expressions, inverses, nominals, and counting. Technical Report INFSYS RR-1843-10-03, TU Vienna, April 2010.

[Ortiz *et al.*, 2010b] M. Ortiz, S. Rudolph, and M. Šimkus. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In *Proc. KR'10*, 2010.

[OWL Working Group, 2009] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009.

[Rudolph and Glimm, 2010] S. Rudolph and B. Glimm. Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *J. Artif. Intell. Res. (JAIR)*, 39:429–481, 2010.

[Tobies, 2000] S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. Artif. Intell. Res. (JAIR)*, 12:199–217, 2000.

[Vardi, 1998] Moshe Y. Vardi. Reasoning about the past with two-way automata. In *ICALP'98*, pages 628–641. Springer, 1998.