First Order Rewritability for Ontology Mediated Querying in Horn- \mathcal{DLFD}

David Toman and Grant Weddell

Cheriton School of CS, University of Waterloo, Canada {david,gweddel}@uwaterloo.ca

Abstract. We study first order (FO) rewritability for query answering in the setting of ontology mediated queries (OMQ) in which ontologies are formulated in the Horn fragments of the feature description logic \mathcal{DLFD} . In general, OMQ approaches for logics such as \mathcal{DLFD} rely on non-FO completion of the data (ABoxes) that can commonly be expressed as Datalog programs. In this paper, we study the problem of the existence of FO rewritings for a given Horn- \mathcal{DLFD} theory (TBox) in terms of *Beth definability*, and show how *Craig interpolation* can then be used to effectively construct the rewritings (when they exist) from the *Clark's completion* of Datalog programs for computing ABox completions. In addition, since data that populates OMQ is commonly residing in relational database systems, we show how constraints *enforced* by such systems can be used to expand the scope of rewritable queries.

1 Introduction

In earlier work [25,27], we presented a combined-combined approach to ontology mediated querying (OMQ) of backend relational data sources when ontologies are formulated as TBoxes in the Horn fragments of the *description logic* (DL) \mathcal{DLFD} and where ABoxes are induced by data residing in the data sources. The logic \mathcal{DLFD} is the most expressive member of the FunDL family of feature-based DLs [26], which are particularly well suited for capturing relational schemata by virtue of being feature-based and also by virtue of including the path functional dependency (PFD) concept constructor. PFDs makes it possible to capture a variety of equality generating dependencies commonly occurring in relational schemata, such as primary keys, uniqueness constraints and functional dependencies. To illustrate, we show in the next section how the relational schema in Figure 1 can be captured as a Horn- \mathcal{DLFD} TBox, the FunDL dialect that will be the focus of the remainder of this paper. This example also serves to illustrate how the parts of a TBox that reflect a relational schema could in principle be automatically generated, and thereby make it unnecessary to author so-called mapping rules in *ontology based data access* (OBDA) (again see [25,27]).

The combined-combined approach is so named because any of the Horn fragments of \mathcal{DLFD} in the FunDL family will require using ontological knowledge,

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

a Horn- \mathcal{DLFD} TBox in our case, for two steps in OMQ. The first step, as with the *combined approach* to OMQ [19,20,22,23], is to complete the ABox via a Datalog program that adds tuples to tables in the backend relational database. Note that this first step depends only on the TBox. The second step, as with the *perfect rewriting approach* to OMQ and OBDA [9], is to rewrite a query to account for such artifacts as inheritance and attribute typing in the TBox.

In this paper, we show how *Beth definability* can be used to resolve the problem of recognizing when a first order (FO) rewriting of the Datalog program needed in the first step exists, and then how *Craig interpolation* can be used to effectively construct such a rewriting. A key insight is to employ the *Clark's completion* of the Datalog program in both the diagnosis and synthesis of FO expressibility of ABox completion.

The existence of such a rewriting enables an OMQ frontend to a relational data source to operate entirely by a more refined query reformulation that, for a given user query, ultimately yields an SQL query over the data source, that is, with no requirement to update tables in the data source beforehand. An additional advantage of our approach is that it also becomes straightforward to incorporate the benefits of integrity constraints such as foreign key constraints that are enforced by a relational database management system to simplify query reformulation. Indeed, such integrity constraints can enable FO rewriting of data completion that would otherwise not be possible.

In summary, our contributions are as follows.

- We show how to decide FO rewritability of OMQ in Horn-DLFD via Clark's completion of Datalog programs and Beth definability;
- 2. We show how an equivalent non-recursive Datalog program can be derived via Craig's theorem and interpolation in cases where ABox completion for a given TBox is indeed FO expressible;
- 3. We illustrate how integrity constraints in backend relational schema can enable FO rewritability of OMQ that would otherwise not be possible; and
- 4. We show how our framework extends to *query specific OMQ* in a straightforward manner.

The remainder of the paper is organized as follows. Section 2 immediately following provides the necessary background and definitions. Here, we review Horn- \mathcal{DLFD} and the combined combined approach to OMQ. Our main results follow in Section 3 in which we show how the above-mentioned artifacts, Clark's completion of Datalog programs for example, can be employed to both decide FO rewritability and to synthesize FO rewritings of ABox completion in the combined combined approach to OMQ. In Section 4 shows how a Datalog program computing an ABox completion can usefully incorporate database constraints enforced by backend relational database systems. Section 5 then considers query specific OMQ, that is, when the combination of a Datalog ABox completion procedure and a *specific* query are FO expressible. Summary comments and our conclusions are in Section 6.

```
create table EMP ( name STRING not null,
  dept STRING, man STRING,
  primary key ( name ),
  constraint deptRef foreign key ( dept ) references DEPT,
  constraint manRef foreign key( man ) references EMP );
create table DEPT ( name STRING not null,
  head STRING,
  primary key ( name ),
  candidate key ( head ),
  constraint headRef foreign key( head ) references EMP );
```

Fig. 1. A Relational Schema.

2 Background and Definitions

All member dialects of the FunDL family of DLs [26] are fragments of FOL with underlying signatures based on disjoint sets of unary predicate symbols PC, called *primitive concepts*, constant symbols IN, called *individuals*, and unary function symbols F, called *features*. The latter replace *roles* in other DLs and are interpreted as either total functions or partial functions, depending on the particular FunDL dialect.

The most expressive member of the FunDL family is the dialect \mathcal{DLFD} . In this paper, we consider Horn- \mathcal{DLFD} , the most expressive Horn fragment of \mathcal{DLFD} , assuming in particular that an ontology is given as a Horn- \mathcal{DLFD} TBox.¹ To simplify our presentation, we also assume features are interpreted as total functions.

Definition 1 (Horn- \mathcal{DLFD}) A path function Pf is a word in F^{*}, with the empty word denoted by id, and concatenation by ".". Concept descriptions of two kinds, C and D, are defined by the grammar rules on the left-hand-side of Figure 2, where $A \in PC$, $f \in F$ and Pf (possibly subscripted) refers to a path function. An instance of the final production is called a path functional dependency (*PFD*).

Semantics is defined in the standard way with respect to an interpretation $\mathcal{I} = (\triangle, (\cdot)^{\mathcal{I}})$ that fixes the meaning of symbols in PC, IN and F. Here, features are interpreted as total functions. We also assume the unique name assumption (UNA) whereby, for any distinct individuals a and b, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

The interpretation function \mathcal{I} is extended to path expressions by interpreting id (the empty word) as the identity function $\lambda x.x$, concatenation as function composition, and to complex concept descriptions C or D as given on the righthand-side of Figure 2. An interpretation \mathcal{I} satisfies a subsumption $C \sqsubseteq D$ if

¹ Here, we do not consider the problem of recognizing when TBoxes in very expressive DLs have unique minimal models.

Syntax SEMANTICS: " $(\cdot)^{\mathcal{I}}$ " $A^{\mathcal{I}} \subseteq \triangle$ C ::= A $\{x \mid \mathsf{Pf}^{\mathcal{I}}(x) \in C^{\mathcal{I}}\}$ $\mid \forall \mathsf{Pf.}C$ $C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$ $\mid C_1 \sqcap C_2$ $C^{\mathcal{I}} \subset \bigtriangleup$ D ::= C| _ $\{x \mid \exists y \in \triangle : f^{\mathcal{I}}(y) = x\}$ $| \exists f^{-1}$ $\{x \mid \mathsf{Pf}^{\mathcal{I}}(x) \in D^{\mathcal{I}}\}$ $| \forall \mathsf{Pf}.D$ $| C: \mathsf{Pf}_1, \dots, \mathsf{Pf}_k \to \mathsf{Pf} \ \{x \mid \forall y \in C^{\mathcal{I}}: \bigwedge_{i=1}^k \mathsf{Pf}_i^{\mathcal{I}}(x) = \mathsf{Pf}_i^{\mathcal{I}}(y) \Rightarrow \mathsf{Pf}^{\mathcal{I}}(x) = \mathsf{Pf}^{\mathcal{I}}(y) \}$

Fig. 2. Horn- \mathcal{DLFD} Concepts.

 $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, a concept assertion A(a) if $a^{\mathcal{I}} \in A^{\mathcal{I}}$, and a feature assertion f(a) = b if $f^{\mathcal{I}}(a^{\mathcal{I}}) = b^{\mathcal{I}}$, where $a, b \in \mathsf{IN}$.

A knowledge base (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox \mathcal{T} of subsumptions, and an ABox \mathcal{A} of assertions. \mathcal{I} satisfies \mathcal{K} if it satisfies each subsumption and assertion in \mathcal{K} .

Although incorporating features deviates from the normal practice of using binary predicate symbols called *roles*, features make it easier to incorporate concept constructors suited to the capture of relational data sources that include various dependencies by a straightforward reification of *n*-ary predicates. Thus, e.g., a role *R* can be reified as a primitive concept R_C and two features *domR* and *ranR* in Horn- \mathcal{DLFD} , and a subsumption $A \sqsubseteq \forall R.B$ can then be captured as a subsumption $\forall domR.A \sqsubseteq \forall ranR.B$.

As usual, to ensure decidability in the presence of ABoxes [28] and reasonable computational properties of the logic, one looks for additional restrictions on the PFD concept constructors. One restriction, which has been investigated (e.g., [17,27]), is to limit the PFD constructor to one of the following two forms:

1.
$$C : \mathsf{Pf}_1, \dots, \mathsf{Pf} . \mathsf{Pf}_i, \dots, \mathsf{Pf}_k \to \mathsf{Pf}$$
 or
2. $C : f_1, \dots, f_k \to g$

We can allow a more complex variant of (2) if the ABox completion admits/requires creating additional constant symbols; for details—beyond the scope of this paper—see [27]. Also, in this paper we will not explore the parameter tractable fragments of Horn- \mathcal{DLFD} nor the issues connected with interpreting features as partial functions. Both these issues were studied in [25] which shows, among other things, how partiality can be easily simulated in Horn- \mathcal{DLFD} .

Example 2 Figure 3 lists the subsumptions for a Horn- \mathcal{DLFD} TBox that provides an ontological view of the relational schema illustrated in Figure 1. Part (a) are subsumptions that capture column definitions and data dependencies manifest in the schema itself. This part of an ontology can be automatically generated

```
(disjoint classes) EMP \sqcap DEPT \sqsubseteq \bot
                                                                           (disjoint columns) \text{EMP} \sqsubseteq \text{DEPT} : \text{name} \rightarrow id
                               \texttt{EMP} \sqcap \texttt{STRING} \sqsubseteq \bot
                                                                               (b) ADDITIONAL DATA DEPENDENCIES
                               \texttt{DEPT} \sqcap \texttt{STRING} \sqsubseteq \bot
(attribute typing) \text{EMP} \sqsubseteq \forall \text{name.STRING}
                                                                                   (inheritance) MAN \Box EMP
                               \texttt{EMP} \sqsubseteq \forall \texttt{deptRef.DEPT}
                                                                                  (local typing) \text{EMP} \sqsubseteq \forall \texttt{manRef.MAN}
                                                                                                          DEPT \Box \forall headRef.EMP
                               EMP \square \forall manRef.EMP
                                                                                         (inverses) MAN \square \exists headRef^{-1}
                               \texttt{DEPT} \sqsubseteq \forall \texttt{headRef.MAN}
                               \texttt{DEPT} \sqsubseteq \forall \texttt{name.STRING}
                                                                                     (c) ONTOLOGY ENHANCEMENT
     (primary keys) \text{EMP} \sqsubseteq \text{EMP} : name \rightarrow id
                               \texttt{DEPT} \sqsubseteq \texttt{DEPT} : \texttt{name} \to id
  (candidate keys) DEPT \sqsubseteq DEPT : head \rightarrow name
(a) DERIVED FROM THE RELATIONAL SCHEMA
```

Fig. 3. An Ontology as a Horn- \mathcal{DLFD} TBox.

(for an outline of a procedure to do this, see [25]). Observe in particular how the name attached to a foreign key constraint, such as headRef, is mapped to a feature with the same name. Part (b) illustrates how additional data dependencies not expressible without appeal to general assertions in SQL can also be captured in Horn- \mathcal{DLFD} . Here, the (presumed) fact that name values of employees are disjoint from name values of departments is captured by appeal to an *asymmetric* use of PFDs. Finally, Part (c) illustrates how an ontology can be enhanced by adding further subsumptions, in this case relating to the virtual concept of a manager.

Conjunctive queries are, as usual, formed from atomic queries (or *atoms*) of the form "A(x)" and "x.f = y", where x and y are variables, using conjunction and existential quantification. To simplify notation, we conflate conjunctive queries with the set of its constituent atoms and a set of *answer variables*:

Definition 3 (Conjunctive Queries and Certain Answers) Let $A(x_i)$ and $x_{i_1} \cdot f_i = x_{i_2}$ be (first-order) atoms, where A is a primitive concept, f_i is a feature, and \bar{x} and \bar{y} tuples of variables. A conjunctive query (CQ) φ is an expression of the form $\exists \bar{y} \cdot \bigwedge \psi$ with free variables \bar{x} , that is constructed from the above atoms ψ using conjunctions and existential quantification.

Let \mathcal{K} be a Horn- \mathcal{DLFD} knowledge base and φ a CQ. A certain answer to φ over \mathcal{K} is a substitution of constant symbols \bar{a} , $[\bar{x} \mapsto \bar{a}]$, such that $\mathcal{K} \models \varphi[\bar{x} \mapsto \bar{a}]$.

As usual, UCQ stands for a union of CQs. To study the *first-order* rewritability of conjunctive queries over Horn- \mathcal{DLFD} TBoxes, we use the following version of the combined combined approach to OMQ in our setting.

Proposition 4 (Combined Combined Approach to OMQ [25,27,30]) Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a Horn-DLFD knowledge base and φ a conjunctive query. Then there is a UCQ query $\varphi_{\mathcal{T}}$ and a Datalog program $\Pi_{\mathcal{T}}$, both of which can be effectively constructed from \mathcal{T} , such that

$$\mathcal{K} \models \varphi[\bar{x} \mapsto \bar{a}] \iff \Pi_{\mathcal{T}}(\mathcal{A}) \models \varphi_{\mathcal{T}}[\bar{x} \mapsto \bar{a}]$$

for a tuple of constant symbols \bar{a} and where $\Pi_{\mathcal{T}}(\mathcal{A})$ is the minimal model of $\Pi_{\mathcal{T}}$ when evaluated over \mathcal{A} .

The proposition uses a Datalog program $\Pi_{\mathcal{T}}$ to define an ABox completion over which the query $\varphi_{\mathcal{T}}$, the rewriting of the original user query, is evaluated to compute the certain answers. The Proposition also shows that the *data complexity* of OMQ in Horn- \mathcal{DLFD} is complete for PTIME.

3 Classification of Horn- \mathcal{DLFD} TBoxes

To simplify the presentation, we assume that Horn- \mathcal{DLFD} TBoxes are in a normal form given as follows:

Definition 5 (Normal Form of Horn- \mathcal{DLFD} TBoxes)

Let \mathcal{T} be a Horn- \mathcal{DLFD} TBox. We say that \mathcal{T} is in normal form if all subsumptions adhere to one of the following six forms:

1. $A \sqsubseteq \bot$,	$4. \forall f.A \sqsubseteq B,$
$2. A_1 \sqcap A_2 \sqsubseteq B,$	5. $A \sqsubseteq \exists f^{-1}, and$
$3. A \sqsubseteq \forall f.B,$	6. $A \sqsubseteq B : Pf_1, \dots, Pf_k \to Pf.$

It is an easy exercise to convert an arbitrary Horn- \mathcal{DLFD} TBox to its conservative extension in this normal form by introducing additional primitive concepts.

Definition 6 (Datalog Program $\Pi_{\mathcal{T}}$) The Datalog program $\Pi_{\mathcal{T}}$ used in Proposition 4 consists of completion rules obtained by translating subsumptions that are logical consequences of \mathcal{T} . The form of these subsumptions and their translation are given as follows:

$(consequences \ of \ \mathcal{T})$	(completion rule in $\Pi_{\mathcal{T}}$)
$A \sqsubseteq \bot$	$C_{\perp}(x) \leftarrow C_A(x)$
$A_1 \sqcap A_2 \sqsubseteq B$	$C_B(x) \leftarrow C_{A_1}(x), C_{A_2}(x)$
$A \sqsubseteq \forall f.B$	$C_B(x) \leftarrow C_A(y), R_f(y, x)$
$\forall f.A \sqsubseteq B$	$C_B(x) \leftarrow C_A(y), R_f(x, y)$

For every primitive concept B, we introduce an unary EDB predicate $P_B(x)$ together with an additional clause $C_B(x) \leftarrow P_B(x)$ that captures explicit data in an ABox of the form A(a), and an IDB predicate $C_B(x)$ corresponding to the completion of the ABox w.r.t. \mathcal{T} . For features, we use $R_f(x, y)$ as a synonym for ABox assertion f(x) = y to conform with standard Datalog syntax. We also define an auxiliary symbol $C_{\perp}(x)$ that will be used to test for ABox consistency. Note that the subsumptions for unqualified inverse features (5) and PFDs (6) do not contribute to the definition of an ABox completion since they do not generate additional ABox assertions for primitive concepts. Also note that, unlike in the classical combined approach [19,23], our combined combined approach does *not* introduce any new constants in the ABox [27].

To test for *first-order* definability of the completion (i.e., all predicates that stand for the completed ABox instance), we use the following construction:

Definition 7 (Clark's Completion $\Sigma_{\mathcal{T}}$) The Clark's Completion [11] $\Sigma_{\mathcal{T}}$ of $\Pi_{\mathcal{T}}$ is given by a set of formulas

$$C_B(x) \leftrightarrow P_B(x) \lor (\exists y.\alpha_1) \lor \ldots \lor (\exists y.\alpha_n)$$

corresponding to all clauses $C_B(x) \leftarrow \alpha_i \in \Pi_T$ with the head $C_B(x)$.

This completion closes the original Datalog program in the following sense:

Proposition 8 ([11], simplified for this paper)

 $- \Pi_{\mathcal{T}} \models C_B(a) \text{ implies } \Sigma_{\mathcal{T}} \models C_B(a), \text{ and} \\ - \Pi_{\mathcal{T}} \not\models C_B(a) \text{ implies } \Sigma_{\mathcal{T}} \models \neg C_B(a).$

Note that Clark's result works in the much more general setting of logic programs with function symbols and possibly infinite resolution proofs and under the *Negation As Failure* semantics. Since Clark's completion makes all IDB predicates closed, we can now use standard tools for testing for explicit definability. Note that had we used $\Pi_{\mathcal{T}}$ instead, none of the definability results could possibly hold. Note that in absence of role/feature subsumptions (such as role hierarchies) we do not need to apply the completion to the R_f atoms.

Proposition 9 (Projective Beth Definability [3]) Let Σ be an FO theory over symbols in L and $L' \subseteq L$. Then the following are equivalent:

1. For M_1 and M_2 models of Σ such that $M_{1|L'} = M_{2|L'}$, it holds that $M_1 \models \varphi[\mathbf{a}]$ iff $M_2 \models \varphi[\mathbf{a}]$ for all M_1 , M_2 , and \mathbf{a} tuples of constants, and 2. φ is equivalent under Σ to a formula ψ in L'.

This gives us a complete characterization of first-order rewritability of the ABox closure of individual primitive concepts with respect to Horn- \mathcal{DLFD} TBoxes as follows:

Theorem 10 Let \mathcal{T} be a Horn- \mathcal{DLFD} TBox over the primitive concepts and features $\{\mathcal{A}_1, \ldots, \mathcal{A}_k, f_1, \ldots, f_n\}$, and let \mathcal{T}' be a conservative extension of \mathcal{T} in normal form. Then the completion of the primitive concept \mathcal{A} w.r.t. \mathcal{T} is first-order definable if and only if $C_{\mathcal{A}_i}(x)$ is Beth definable over $\Sigma_{\mathcal{T}'}$ and $L' = \{P_{\mathcal{A}_1}, \ldots, P_{\mathcal{A}_k}, R_{f_1}, \ldots, R_{f_n}\}$.

Proof. Follows immediately from the properties of Beth definability (Proposition 9) and the definition and properties of the Clark's completion (Proposition 8).

Observe that one can restrict the alphabet of the ABox (L') to target only ABoxes over restricted signature(s).

Given $\Sigma_{\mathcal{T}}$, one can now reformulate (1) in Proposition 9 as a logical implication problem by making a copy of all formulas of $\Sigma_{\mathcal{T}}$ in which all non-logical symbols not in $\{P_{A_1}, \ldots, P_{A_k}, R_{f_1}, \ldots, R_{f_n}\}$ are starred. Hence, the definability question for $C_A(x)$ can be expressed as a logical implication question of the form:

$$\Sigma_{\mathcal{T}} \cup \Sigma_{\mathcal{T}}^* \models \forall x. C_A(x) \to C_A^*(x) \tag{1}$$

Note that, on closer inspection, all formulas in $\Sigma_{\mathcal{T}}$ can be written as \mathcal{ALCI} subsumptions.² Hence:

Theorem 11 Let \mathcal{T} be a Horn- \mathcal{DLFD} TBox. Then the existence of

- 1. the first order rewritability of the \mathcal{A} completion with respect to \mathcal{T} ,
- 2. the consistency of \mathcal{T} , and
- 3. the uniform query rewritability over \mathcal{T}

are all decidable and in EXPTIME.

Proof. The first claims follow immediately from Theorem 10 applied to all atoms of the form C_B and the decidability and complexity of reasoning in \mathcal{ALCI} . The second claim relies on definability of $C_{\perp}(x)$ and, in the presence of key PFDs of the form $A \sqsubseteq B$: $\mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}$, on the definability of $C_A(x)$ and $C_B(x)$ since the remaining consistency check for PFD satisfaction can then be expressed as a first-order query with respect to these explicit definitions. The last claim follows by observing that (i) definability of atomic queries implies definability of arbitrary (U)CQs using the combined-combined approach [25], and that (ii) non-definability of a single atomic query exhibits the need for a non first-order ABox completion for queries containing/consisting of this atom.

Note that in the third case one can restrict the definability conditions to atoms that can appear in the query $\varphi_{\mathcal{T}}$. A matching lower bound can be obtained for expressive fragments of Horn- \mathcal{DLFD} (for which the reasoning complexity is EXPTIME-complete). However, the exact complexity is open for PTIME fragments, such as \mathcal{CFD}_{nc} and $\mathcal{CFDI}_{kc}^{\forall-}$. Note, however, that since the size (and the construction) of rewritings will dominate this cost (even for the simplest ontology languages [18]), exact complexity bounds are mostly of academic interest.

Example 12 Returning to our introductory example, we find out that neither $C_{\text{EMP}}(x)$ nor $C_{\text{DEPT}}(x)$ nor the TBox-derived $C_{\text{MAN}}(x)$ atom are first-order definable. We confirmed this using a tableau-based depth-limited interpolant generator [16,31]. One of the reasons is that an ABox can contain a chain of alternating deptRef/empRef features without having the objects along this chain alternately belonging to the EMP and DEPT concepts, save the individual at the beginning of this chain. The Datalog completion rules are needed to propagate these concept assertions along such a chain.

 $^{^2}$ Disregarding functionality of the original features does not matter at this juncture.

3.1 Construction of Rewritings

To obtain an algorithm that constructs rewritings from our characterization of FO rewritability, we utilize Craig Interpolation:

Proposition 13 (Craig Interpolation [12]) Let φ and ϕ be first order formulas such that $\models \varphi \rightarrow \phi$. Then there is a first order formula ψ , called Craig interpolant, containing only symbols common to φ and ϕ such that $\models \varphi \rightarrow \psi$ and $\models \psi \rightarrow \phi$.

Moreover, the interpolant can be extracted, typically in linear time, from a proof of $\models \varphi \rightarrow \phi$, as long as a reasonably structural proof system, such as resolution, (cut-free) sequent calculus, and/or analytic tableau is used.

The formulation (1), after a minor manipulation of the formulas, can thus be used to construct a Craig interpolant for $C_A(x)$ from a (analytic) tableau proof of the above logical implication statement (if one exists): one can construct, with the help of blocking [1,13], a finite analytic tableau for (1). The size of the rewriting, however, will depend crucially on the proof system used and on the presentation of the rewriting. However, by using an optimal tableau for $A\mathcal{LC}$ [13] (modified for $A\mathcal{LCT}$), one can construct an exponential *representation* of the rewriting by sharing common subexpressions in the same way the tableau algorithm *caches* already unsatisfiable sets. However, plain first-order representation of the formula will suffer from additional exponential blowup, as one would expect.

Combining the above construction with the already existing rewriting $\varphi_{\mathcal{T}}$ from [25] we get:

Theorem 14 Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a Horn- \mathcal{DLFD} knowledge base. Then the data complexity of conjunctive query answering is in AC^0 whenever the \mathcal{A} completion with respect to \mathcal{T} is first-order definable with respect to $\Sigma_{\mathcal{T}}$.

Proof. Let $\psi_A(x)$ be a first order definition of C_A w.r.t. $\Sigma_{\mathcal{T}}$. Then $\mathcal{K} \models \varphi(\mathbf{a})$ iff $\mathcal{A} \models \varphi_{\mathcal{T}}[\psi_A[y/x]/A(y) \mid A \in \mathsf{PC}](\mathbf{a})$. The claim follows since $\varphi_{\mathcal{T}}[\psi_A[y/x]/A(y) \mid A \in \mathsf{PC}]$ is a first order formula, in particular a UCQ in the case of Horn- \mathcal{DLFD} .

4 Integration of Database Constraints

For ABoxes constructed from relational instances (see the discussion in our Introduction and Background Sections), relational systems commonly enforce *integrity constraints* such as *primary* and *foreign* keys. The effect of these database constraints is that parts of the corresponding ABox are not only *consistent* with such constraints but are a *model*. This observation is in particular helpful in the case of *foreign keys* since they stipulate that parts of the ABox completion are no longer necessary: for a foreign key to hold in an instance of a relational database, either its *target* exists in the appropriate table or the source is NULL, which can then be interpreted as *value unknown* and is taken care of by query rewriting $\varphi_{\mathcal{T}}$. **Definition 15 (Adding Foreign Keys)** Let $A \sqsubseteq B$ and $A \sqsubseteq \forall f.B$ be subsumptions corresponding to foreign keys that capture ISA and attribute typing, respectively. For each such constraint, add $P_B(x) \leftarrow C_A(x)$ and $P_B(x) \leftarrow C_A(y), R_f(x, y)$, respectively, to Σ_A .

This definition allows us to sidestep parts of the ABox completion that are mandated by the database constraints and thus *already exist* in the original instance of the ABox. As a consequence, we have the following extension of Theorem 14:

Theorem 16 Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a Horn- \mathcal{DLFD} knowledge base. Then the data complexity of conjunctive query answering is in AC^0 whenever the \mathcal{A} completion with respect to \mathcal{T} is first-order definable with respect to the theory $\Sigma_{\mathcal{T}} \cup \Sigma_{\mathcal{A}}$.

Example 17 Continuing with our example, adding the constraints

$$\begin{split} P_{\text{EMP}}(x) &\leftarrow C_{\text{EMP}}(y), R_{\text{manRef}}(x, y) \\ P_{\text{EMP}}(x) &\leftarrow C_{\text{DEPT}}(y), R_{\text{deptRef}}(x, y) \\ P_{\text{DEPT}}(x) &\leftarrow C_{\text{EMP}}(y), R_{\text{headRef}}(x, y) \end{split}$$

to $\Sigma_{\mathcal{A}}$, both $C_{\text{EMP}}(x)$ and $C_{\text{DEPT}}(x)$ become definable with the expected interpolants $C_{\text{EMP}}(x)$ and $C_{\text{DEPT}}(x)$ found by the above-mentioned interpolant generator [16,31]. Moreover, $C_{\text{MAN}}(x)$ also becomes definable by $\exists y. P_{\text{EMP}}(x) \wedge R_{\text{manRef}}(y, x)$ (again, using the interpolant found by the interpolant generator).

5 Query-specific Rewritability

Our approach also provides decidability for the non-uniform (query-specific) problems:

Theorem 18 Let \mathcal{T} be a Horn- \mathcal{DLFD} TBox and φ a (U)QC. Then the following are equivalent:

1. $\Sigma_{\mathcal{T}} \cup \Sigma_{\mathcal{T}}^* \models \forall x. \varphi_{\mathcal{T}} \to \varphi_{\mathcal{T}}^*(x), and$ 2. φ is first-order rewritable with respect to \mathcal{T} .

The exact complexity again depends on the complexity of (1) above. In the general case, an EXPTIME bound follows from [15], but again, a more refined analysis is in order for fragments of Horn- \mathcal{DLFD} .

6 Summary and Conclusions

We have shown how the combined combined approach to OMQ can be used to determine when query answering mediated by a Horn- \mathcal{DLFD} TBox can be reduced to a first order query over a plain ABox (or, in turn, over the underlying data source), and when a non-first order (Datalog-based) completion of the data is needed. Interestingly enough, and unlike other approaches that aim on determining first-order rewritability, our approach links the problem to well studied issues of *explicit definability* [3], interpolation [12], and to Clark's completion of logic programs [11]. In addition, our approach naturally and seamlessly incorporates *database constraints* that hold in the data source to improve our chances of finding a first order rewriting.

Further extensions of this work along the following lines is needed:

- Performing a fine-grained analysis of computational complexity for tractable members of Horn- \mathcal{DLFD} , such as \mathcal{CFD}_{nc} . (We conjecture that the rewritability problem will remain intractable.)
- Determining if a dichotomy holds, in particular, if non-rewritability implies NL hardness of OMQ in data complexity. (We conjecture that this is the case since blocked branches of a tableau proof signal non-rewritability and can be used to simulate, e.g., s-t connectivity similarly to the approach in [21].)
- Extensions to situations in which ABox closure requires a finite number of additional *auxiliary* constants. (The approach itself can immediately use Skolem functions to deal with such constants while retaining completeness. However, decidability and complexity issues stemming from the use of Skolem functions remain to be determined.)
- Extensions to DLs outside of the FunDL family, such as the various members of the \mathcal{EL} family and to more expressive logics such as Horn- \mathcal{SHIQ} .
- And integration with an interpolation-based relational query optimizer [16,31] via *depth-limited* tableau construction, which, for a sufficiently large depth, leads to completeness as in [10].

An additional issue arises from the assumption of UNA. Indeed, this assumption is implicit in [21], where explicit equality among constant symbols is simply not considered, and also in most of the other approaches to OMQ via query rewriting, including the original approach introduced in [9]. In the case of Horn- \mathcal{DLFD} , UNA is needed explicitly to accommodate keys and PFDs. We conjecture that some form of UNA is necessary, but whether restricted variants could suffice is unknown.

6.1 Related Work

First order rewritability for Horn logics in the $\mathcal{ALC}/S\mathcal{HIQ}$ family has been studied by others, e.g., see [4,6]. This earlier work has also developed algorithms for generating such rewritings efficiently for logics in the \mathcal{EL} family [14]. Our approach seems to provide an alternative path to detecting rewritability and to generating rewritings. A feature of our approach is its link to interpolation-based query optimization [16,29]. The link to query optimization reveals that minimal sized rewritings are often not optimal for query execution. However, establishing limits on the size of rewriting [5] does provide a guide to what rewritings are reasonable to consider during query optimization. The use of database constraints, or constraints implied by data mapping rules and database constraints, has been explored in several systems that implement variants of perfect rewriting [9], such as Ontop and MASTRO [2,8] and others. Our approach, however, seamlessly accommodates such constraints into the rewriting via interpolation and, again, could serve as an unifying approach between perfect rewriting-based approaches and the combined approach, even for DL-Lite based approaches to OMQ/OBDA.

Our approach cannot deal with rewritings to Datalog/and or Linear Datalog and with dichotomies on the NL-PTIME [21] and PTIME-coNP boundaries [24]: extensions of definability/interpolation beyond first order theories do not seem to exist at present and it is unclear whether similar approaches can be developed due to the close ties between interpolation and compactness of first order theories.

Finally, note that *Beth definability* and *Craig Interpolation* have been used for other purposes, such as query reformulation under first-order constraints [7,16,29,31]. That use, however, is orthogonal to the topic of this paper.

References

- Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation,* and Applications. Cambridge University Press, 2003.
- Timea Bagosi, Diego Calvanese, Josef Hardi, Sarah Komla-Ebri, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, Mindaugas Slusnys, and Guohui Xiao. The ontop framework for ontology based data access. In *The Semantic Web and Web Science - 8th Chinese Conference, CSWS 2014, Revised Selected Papers*, pages 67–77, 2014.
- 3. Evert Willem Beth. On Padoa's method in the theory of definition. *Indagationes Mathematicae*, 15:330–339, 1953.
- 4. Meghyn Bienvenu, Peter Hansen, Carsten Lutz, and Frank Wolter. First orderrewritability and containment of conjunctive queries in horn description logics. In *Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa*, 2016.
- Meghyn Bienvenu, Stanislav Kikot, Roman Kontchakov, Vladimir V. Podolskii, Vladislav Ryzhikov, and Michael Zakharyaschev. The complexity of ontologybased data access with OWL 2 QL and bounded treewidth queries. In Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, pages 201–216, 2017.
- Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontologybased data access: A study through disjunctive datalog, csp, and MMSNP. ACM Trans. Database Syst., 39(4):33:1–33:44, 2014.
- Alexander Borgida, Jos de Bruijn, Enrico Franconi, Inanç Seylan, Umberto Straccia, David Toman, and Grant E. Weddell. On finding query rewritings under expressive constraints. In SEBD, pages 426–437, 2010.
- Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The MASTRO system for ontology-based data access. Semantic Web, 2(1):43–53, 2011.

- Diego Calvanese, Giuseppe de Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. J. of Automated Reasoning, 39(3):385–429, 2007.
- Jan Chomicki. Depth-bounded bottom-up evaluation of logic program. J. Log. Program., 25(1):1–31, 1995.
- Keith L. Clark. Negation as failure. In Logic and Data Bases, Symposium on Logic and Data Bases, Centre d'études et de recherches de Toulouse, France, 1977, pages 293–322, 1977.
- 12. William Craig. Three uses of the Herbrand-Genzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, 22:269–285, 1957.
- Francesco M. Donini and Fabio Massacci. EXPTIME tableaux for ALC. Artif. Intell., 124(1):87–138, 2000.
- Peter Hansen, Carsten Lutz, Inanç Seylan, and Frank Wolter. Efficient query rewriting in the description logic EL and beyond. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, pages 3034–3040, 2015.
- Ian Horrocks, Ulrike Sattler, Sergio Tessaris, and Stephan Tobies. How to decide query containment under constraints using a description logic. In Logic for Programming and Automated Reasoning, 7th International Conference, LPAR 2000, Reunion Island, France, November 11-12, 2000, Proceedings, pages 326–343, 2000.
- Alexander K. Hudek, David Toman, and Grant E. Weddell. On enumerating query plans using analytic tableau. In Automated Reasoning with Analytic Tableaux and Related Methods - 24th International Conference, TABLEAUX 2015, Wrocław, Poland, September 21-24, 2015. Proceedings, pages 339–354, 2015.
- Vitaliy L. Khizder, David Toman, and Grant Weddell. Reasoning about Duplicate Elimination with Description Logic. In DOOD, pages 1017–1032, 2000.
- Stanislav Kikot, Roman Kontchakov, Vladimir V. Podolskii, and Michael Zakharyaschev. Long rewritings, short rewritings. In Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012, 2012.
- Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in DL-Lite. In Proc. KR, pages 247–257, 2010.
- Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI), pages 2656–2661, 2011.
- Carsten Lutz and Leif Sabellek. Ontology-mediated querying with the description logic EL: trichotomy and linear datalog rewritability. In *Proceedings of the Twenty-*Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, pages 1181–1187, 2017.
- 22. Carsten Lutz, Inanç Seylan, David Toman, and Frank Wolter. The combined approach to OBDA: Taming role hierarchies using filters. In *ISWC (1)*, pages 314–330, 2013.
- Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic *EL* using a relational database system. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 2070–2075, 2009.
- 24. Carsten Lutz and Frank Wolter. Non-uniform data complexity of query answering in description logics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012*, 2012.

- Stephanie McIntyre, Alexander Borgida, David Toman, and Grant E. Weddell. On limited conjunctions and partial features in parameter-tractable feature logics. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 2995–3002, 2019.
- 26. Stephanie McIntyre, David Toman, and Grant E. Weddell. FunDL A family of feature-based description logics, with applications in querying structured data sources. In *Description Logic, Theory Combination, and All That Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*, pages 404–430, 2019.
- 27. Jason St. Jacques, David Toman, and Grant E. Weddell. Object-relational queries over \mathcal{CFDI}_{nc} knowledge bases: OBDA for the SQL-Literate. In *Proc. IJCAI*, pages 1258–1264, 2016.
- 28. David Toman and Grant E. Weddell. On keys and functional dependencies as first-class citizens in description logics. J. Aut. Reasoning, 40(2-3):117–132, 2008.
- David Toman and Grant E. Weddell. Fundamentals of Physical Design and Query Compilation. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.
- 30. David Toman and Grant E. Weddell. Conjunctive Query Answering in CFD_{nc}: A PTIME Description Logic with Functional Constraints and Disjointness. In Australasian Conference on Artificial Intelligence, pages 350–361, 2013.
- 31. David Toman and Grant E. Weddell. An interpolation-based compiler and optimizer for relational queries (system design report). In *IWIL@LPAR 2017 Workshop* and LPAR-21 Short Presentations, Maun, Botswana, May 7-12, 2017, 2017.