How to decide Query Containment under Constraints using a Description Logic

Ian Horrocks[1], Ulrike Sattler[2], Sergio Tessaris[1], and Stephan Tobies[2]

[1] Department of Computer Science, University of Manchester, UK
[2] LuFg Theoretical Computer Science, RWTH Aachen, Germany

Abstract. We present a procedure for deciding (database) query containment under constraints. The technique is to extend the logic $\mathcal{DLR}$ with an Abox, and to transform query subsumption problems into $\mathcal{DLR}$ Abox satisfiability problems. Such problems can then be decided, via a reification transformation, using a highly optimised reasoner for the $\mathcal{SHIQ}$ description logic. We use a simple example to support our hypothesis that this procedure will work well with realistic problems.

1 Introduction

Query containment under constraints is the problem of determining whether the result of one query is contained in the result of another query for every database satisfying a given set of constraints (derived, for example, from a schema). This problem is of particular importance in information integration (see [10]) and data warehousing where, in addition to the constraints derived from the source schemas and the global schema, inter-schema constraints can be used to specify relationships between objects in different schemas (see [6]).

In [12], query containment without constraints was shown to be NP-complete, and a subsequent analysis identified cycles in queries as the main source of complexity [13]. Query containment under different forms of constraints have, e.g., been studied in [23] (containment w.r.t. functional and inclusion dependencies) and [11, 24] (containment w.r.t. *is-a* hierarchies).

Calvanese et al. [4] have established a theoretical framework using the logic $\mathcal{DLR}$,[1] presented several (un)decidability results, and described a method for solving the decidable cases using an embedding in the propositional dynamic logic CPDL$_g$ [17, 15]. The importance of this framework is due to the high expressive power of $\mathcal{DLR}$, which allows Extended Entity-Relationship (EER) schemas and inter-schema constraints to be captured. However, the embedding technique does not lead directly to a practical decision procedure as there is no (known) implementation of a CPDL$_g$ reasoner. Moreover, even if such an implementation were to exist, similar embedding techniques [14] have resulted in severe tractability problems when used, for example, to embed the $\mathcal{SHIF}$ description logic in $\mathcal{SHF}$ by eliminating inverse roles [18].

[1] Set semantics is assumed in this framework.

In this paper we present a practical decision procedure for the case where neither the queries nor the constraints contain regular expressions. This represents a restriction with respect to the framework described in Calvanese et al., where it was shown that the problem is still decidable if regular expressions are allowed in the schema and the (possibly) containing query, but this seems to be acceptable when modelling classical relational information systems, where regular expressions are seldom used [7, 6]. When excluding regular expressions, constraints imposed by EER schemas can still be captured, so the restriction (to contain no regular expressions) is only relevant to inter-schema constraints. Hence, the use of $\mathcal{DLR}$ in both schema and queries still allows for relatively expressive queries, and by staying within a strictly first order setting we are able to use a decision procedure that has demonstrated good empirical tractability.

The procedure is based on the method described by Calvanese et al., but extends $\mathcal{DLR}$ by defining an *ABox*, a set of axioms that assert facts about named *individuals* and tuples of named individuals (see [5]). This leads to a much more natural encoding of queries (there is a direct correspondence between variables and individuals), and allows the problem to be reduced to that of determining the satisfiability of a $\mathcal{DLR}$ *knowledge base* (KB), i.e., a combined schema and ABox. This problem can in turn be reduced to a KB satisfiability problem in the $\mathcal{SHIQ}$ description logic, with n-ary relations reduced to binary ones by reification. In [24], a similar approach is presented. However, the underlying description logic ($\mathcal{ALCNR}$) is less expressive than $\mathcal{DLR}$ and $\mathcal{SHIQ}$ (for example, it is not able to capture Entity-Relationship schemas).

We have good reasons to believe that this approach represents a practical solution. In the FaCT system [18], we already have an (optimised) implementation of the decision procedure for $\mathcal{SHIQ}$ schema satisfiability described in [21], and using FaCT we have been able to reason very efficiently with a realistic schema derived from the integration of several Extended Entity-Relationship schemas using $\mathcal{DLR}$ inter-schema constraints (the schemas and constraints were taken from a case study undertaken as part of the Esprit DWQ project [7, 6]). In Section 4, we use the FaCT system to demonstrate the empirical tractability of a simple query containment problem with respect to the integrated DWQ schema. FaCT's schema satisfiability algorithm can be straightforwardly extended to deal with ABox axioms (and thus arbitrary query containment problems) [22], and as the number of individuals generated by the encoding of realistic query containment problems will be relatively small, this extension should not compromise empirical tractability.

Most proofs are either omitted or given only as outlines in this paper. For full details, please refer to [20] .

2 Preliminaries

In this section we will (briefly) define the key components of our framework, namely the logic $\mathcal{DLR}$, (conjunctive) queries, and the logic $\mathcal{SHIQ}$.

2.1 The Logic $\mathcal{DLR}$

We will begin with $\mathcal{DLR}$ as it is used in the definition of both schemas and queries. $\mathcal{DLR}$ is a description logic (DL) extended with the ability to describe relations of any arity. It was first introduced in [9].

Definition 1. *Given a set of atomic concept names* NC *and a set of atomic relation names* NR*, every* $C \in \mathsf{NC}$ *is a concept and every* $\boldsymbol{R} \in \mathsf{NR}$ *is a relation, with every* $\boldsymbol{R}$ *having an associated arity. If* C, D *are concepts,* $\boldsymbol{R}, \boldsymbol{S}$ *are relations of arity* n*,* i *is an integer* $1 \leqslant i \leqslant n$*, and* k *is a non-negative integer, then*

$$\begin{array}{ll} \top,\ \neg C,\ C \sqcap D,\ \exists[\$i]\boldsymbol{R},\ (\leq k[\$i]\boldsymbol{R}) & \text{are } \mathcal{DLR} \text{ concepts, and} \\ \top_n,\ \neg\boldsymbol{R},\ \boldsymbol{R} \sqcap \boldsymbol{S},\ (\$i/n : C) & \text{are } \mathcal{DLR} \text{ relations with arity } n. \end{array}$$

Relation expressions must be well-typed in the sense that only relations with the same arity can be conjoined, and in constructs like $\exists[\$i]\boldsymbol{R}$ *the value of* i *must be less than or equal to the arity of* $\boldsymbol{R}$*.*

The semantics of $\mathcal{DLR}$ *is given in terms of* interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$*, where* $\Delta^{\mathcal{I}}$ *is the domain (a non-empty set), and* $\cdot^{\mathcal{I}}$ *is an interpretation function that maps every concept to a subset of* $\Delta^{\mathcal{I}}$ *and every* n*-ary relation to a subset of* $(\Delta^{\mathcal{I}})^n$ *such that the following equations are satisfied ("*$\sharp$*" denotes set cardinality).*

$$\begin{array}{rlrl} \top^{\mathcal{I}} = \Delta^{\mathcal{I}} & & (C \sqcap D)^{\mathcal{I}} = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ \neg C^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} & & (\exists[\$i]\boldsymbol{R})^{\mathcal{I}} = & \{d \in \Delta^{\mathcal{I}} \mid \exists (d_1, \ldots, d_n) \in \boldsymbol{R}^{\mathcal{I}}.d_i = d\} \\ & & (\leq k[\$i]\boldsymbol{R})^{\mathcal{I}} = & \{d \in \Delta^{\mathcal{I}} \mid \sharp\{(d_1, \ldots, d_n) \in \boldsymbol{R}^{\mathcal{I}}.d_i = d\} \leq k\} \\ \top_n{}^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n & & \boldsymbol{R}^{\mathcal{I}} \subseteq & \top_n{}^{\mathcal{I}} \\ (\neg\boldsymbol{R})^{\mathcal{I}} = \top_n{}^{\mathcal{I}} \setminus \boldsymbol{R}^{\mathcal{I}} & & (\boldsymbol{R} \sqcap \boldsymbol{S})^{\mathcal{I}} = & \boldsymbol{R}^{\mathcal{I}} \cap \boldsymbol{S}^{\mathcal{I}} \\ & & (\$i/n : C)^{\mathcal{I}} = & \{(d_1, \ldots, d_n) \in \top_n{}^{\mathcal{I}} \mid d_i \in C^{\mathcal{I}}\} \end{array}$$

Note that $\top_n$ does not need to be interpreted as the set of all tuples of arity n, but only as a subset of them, and that the negation of a relation **R** with arity n is relative to $\top_n$.

In our framework, a schema consists of a set of logical inclusion axioms expressed in $\mathcal{DLR}$. These axioms could be derived from the translation into $\mathcal{DLR}$ of schemas expressed in some other data modelling formalism (such as Entity-Relationship modelling [3, 8]), or could directly stem from the use of $\mathcal{DLR}$ to express, for example, inter-schema constraints to be used in data warehousing, (see [6]).

Definition 2. *A* $\mathcal{DLR}$ schema $\mathcal{S}$ *is a set of* axioms *of the form* $C \sqsubseteq D$ *and* $\boldsymbol{R} \sqsubseteq \boldsymbol{S}$*, where* C, D *are* $\mathcal{DLR}$ *concepts and* $\boldsymbol{R}, \boldsymbol{S}$ *are* $\mathcal{DLR}$ *relations of the same arity; an interpretation* $\mathcal{I}$ satisfies $C \sqsubseteq D$ *iff* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$*, and it satisfies* $\boldsymbol{R} \sqsubseteq \boldsymbol{S}$ *iff* $\boldsymbol{R}^{\mathcal{I}} \subseteq \boldsymbol{S}^{\mathcal{I}}$*. An interpretation* $\mathcal{I}$ satisfies *a schema* $\mathcal{S}$ *iff* $\mathcal{I}$ *satisfies every axiom in* $\mathcal{S}$*.*

Crucially, we extend $\mathcal{DLR}$ to assert properties of *individuals*, names representing single elements of the domain. An *ABox* is a set of axioms asserting facts about individuals and tuples of individuals.

Definition 3. *Given a set of individuals* NI*, a* $\mathcal{DLR}$ ABox $\mathcal{A}$ *is a set of* axioms *of the form* $w{:}C$ *and* $\boldsymbol{w}{:}\boldsymbol{R}$*, where* C *is a concept,* $\boldsymbol{R}$ *is a relation of arity* n*,* w *is an individual and* $\boldsymbol{w}$ *is an* n*-tuple* $\langle w_1, \dots, w_n\rangle$ *such that* $w_1, \dots, w_n$ *are individuals. We will often write* w_i *to refer to the* i*th element of an* n*-tuple* $\boldsymbol{w}$*, where* $1 \leqslant i \leqslant n$*.*

Additionally, the interpretation function $\cdot^{\mathcal{I}}$ *maps every individual to an element of* $\Delta^{\mathcal{I}}$ *and thus also tuples of individuals to tuples of elements of* $\Delta^{\mathcal{I}}$*. An interpretation* $\mathcal{I}$ satisfies *an axiom* $w{:}C$ *iff* $w^{\mathcal{I}} \in C^{\mathcal{I}}$*, and it* satisfies *an axiom* $\boldsymbol{w}{:}\boldsymbol{R}$ *iff* $\boldsymbol{w}^{\mathcal{I}} \in \boldsymbol{R}^{\mathcal{I}}$*. An interpretation* $\mathcal{I}$ satisfies *an ABox* $\mathcal{A}$ *iff* $\mathcal{I}$ *satisfies every axiom in* $\mathcal{A}$*.*

A knowledge base *(KB)* $\mathcal{K}$ *is a pair* $\langle \mathcal{S}, \mathcal{A}\rangle$*, where* $\mathcal{S}$ *is a schema and* $\mathcal{A}$ *is an ABox. An interpretation* $\mathcal{I}$ satisfies *a KB* $\mathcal{K}$ *iff it satisfies both* $\mathcal{S}$ *and* $\mathcal{A}$*.*

If an interpretation $\mathcal{I}$ *satisfies a concept, axiom, schema, or ABox* X*, then we say that* $\mathcal{I}$ *is a* model *of* X*, call* X satisfiable*, and write* $\mathcal{I} \models X$*.*

Note that it is not assumed that individuals with different names are mapped to different elements in the domain (the so-called unique name assumption).

Definition 4. *If* $\mathcal{K}$ *is a KB,* $\mathcal{I}$ *is a model of* $\mathcal{K}$*, and* $\mathcal{A}$ *is an ABox, then* $\mathcal{I}'$ *is called an* extension *of* $\mathcal{I}$ *to* $\mathcal{A}$ *iff* $\mathcal{I}'$ *satisfies* $\mathcal{A}$*,* $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$*, and all concepts, relations, and individuals occuring in* $\mathcal{K}$ *are interpreted identically by* $\mathcal{I}$ *and* $\mathcal{I}'$*.*

Given two ABoxes $\mathcal{A}, \mathcal{A}'$ *and a schema* $\mathcal{S}$*,* $\mathcal{A}$ *is* included *in* $\mathcal{A}'$ *w.r.t.* $\mathcal{S}$ *(written* $\langle \mathcal{S}, \mathcal{A}\rangle \models\!\!\approx \mathcal{A}'$*) iff every model* $\mathcal{I}$ *of* $\langle \mathcal{S}, \mathcal{A}\rangle$ *can be extended to* $\mathcal{A}'$*.*

2.2 Queries

In this paper we will focus on conjunctive queries (see [1, chap. 4]), and describe only briefly (in Section 5) how the technique can be extended to deal with disjunctions of conjunctive queries (for full details please refer to [20]). A *conjunctive query* q is an expression

$$q(\boldsymbol{x}) \leftarrow term_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c}) \wedge \ldots \wedge term_n(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c})$$

where $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{c}$ are tuples of *distinguished* variables, variables, and constants, respectively (distinguished variables appear in the answer, "ordinary" variables are used only in the query expression, and constants are fixed values). Each term $term_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c})$ is called an atom in q and is in one of the forms $C(w)$ or $\mathbf{R}(\boldsymbol{w})$, where w (resp. $\boldsymbol{w}$) is a variable or constant (resp. tuple of variables and constants) in $\boldsymbol{x}$, $\boldsymbol{y}$ or $\boldsymbol{c}$, C is a $\mathcal{DLR}$ concept, and $\mathbf{R}$ is a $\mathcal{DLR}$ relation.[2]

For example, a query designed to return the bus number of the city buses travelling in both directions between two stops is:

$$\text{BUS}(nr) \leftarrow \text{bus\_route}(nr, stop_1, stop_2) \wedge \text{bus\_route}(nr, stop_2, stop_1) \wedge \text{city\_bus}(nr)$$

where nr is a distinguished variable (it appears in the answer), $stop_1$ and $stop_2$ are non-distinguished variables, city_bus is a $\mathcal{DLR}$ concept and bus_route is a $\mathcal{DLR}$ relation.

[2] The fact that these concepts and relations can also appear in the schema is one of the distinguishing features of this approach.

In this framework, the *evaluation* $q(\mathcal{I})$ of a query q with n distinguished variables w.r.t. a $\mathcal{DLR}$ interpretation $\mathcal{I}$ (here perceived as standard FO interpretation) is the set of n-tuples $\boldsymbol{d} \in (\Delta^{\mathcal{I}})^n$ such that

$$\mathcal{I} \models \exists \boldsymbol{y}.term_1(\boldsymbol{d}, \boldsymbol{y}, \boldsymbol{c}) \wedge \ldots \wedge term_n(\boldsymbol{d}, \boldsymbol{y}, \boldsymbol{c}).$$

As usual, we require unique interpretation of constants, i.e., in the following we will only consider those intepretations $\mathcal{I}$ with $c^{\mathcal{I}} \neq d^{\mathcal{I}}$ for any two constants $c \neq d$. A query $q(\boldsymbol{x})$ is called *satisfiable* w.r.t a schema $\mathcal{S}$ iff there is an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{S}$ and $q(\mathcal{I}) \neq \emptyset$. A query $q_1(\boldsymbol{x})$ is *contained* in a query $q_2(\boldsymbol{x})$ w.r.t. a schema $\mathcal{S}$ (written $\mathcal{S} \models q_1 \sqsubseteq q_2$), iff, for every model $\mathcal{I}$ of $\mathcal{S}$, $q_1(\mathcal{I}) \subseteq q_2(\mathcal{I})$. Two queries q_1, q_2 are called *equivalent* w.r.t. $\mathcal{S}$ iff $\mathcal{S} \models q_1 \sqsubseteq q_2$ and $\mathcal{S} \models q_2 \sqsubseteq q_1$.

For example, the schema containing the axioms

$$\begin{aligned}(\text{bus\_route} \sqcap (\$1/3 : \text{city\_bus})) &\sqsubseteq \text{city\_bus\_route}\\ \text{city\_bus\_route} &\sqsubseteq (\text{bus\_route} \sqcap (\$1/3 : \text{city\_bus})),\end{aligned}$$

states that the relation city_bus_route contains exactly the bus_route information that concerns city buses. It is easy to see that the following CITY_BUS query

$$\text{CITY\_BUS}(nr) \leftarrow \text{city\_bus\_route}(nr, stop_1, stop_2) \wedge \text{city\_bus\_route}(nr, stop_2, stop_1)$$

is equivalent to the previous BUS query w.r.t. the given schema. In an information integration scenario, for example, this could be exploited by reformulating the BUS query as a CITY_BUS query ranging over a smaller database without any loss of information.

2.3 The Logic $\mathcal{SHIQ}$

$\mathcal{SHIQ}$ is a standard DL, in the sense that it deals with concepts and (only) binary relations (called *roles*), but it is unusually expressive in that it supports reasoning with inverse roles, qualifying number restrictions on roles, transitive roles, and role inclusion axioms.

Definition 5. *Given a set of atomic concept names* NC *and a set of atomic role names* NR *with transitive role names* $\mathsf{NR}_+ \subseteq \mathsf{NR}$*, every* $C \in \mathsf{NC}$ *is a concept, every* $R \in \mathsf{NR}$ *is a role, and every* $R \in \mathsf{NR}_+$ *is a transitive role. If* R *is a role, then* R^- *is also a role (and if* $R \in \mathsf{NR}_+$ *then* R^- *is also a transitive role). If* S *is a (possibly inverse) role,* C, D *are concepts, and* k *is a non-negative integer, then*

$$\top,\ \neg C,\ C \sqcap D,\ \exists S.C,\ \leqslant k S.C \text{ are also } \mathcal{SHIQ} \text{ concepts.}$$

The semantics of $\mathcal{SHIQ}$ *is given in terms of* interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$*, where* $\Delta^{\mathcal{I}}$ *is the domain (a non-empty set), and* $\cdot^{\mathcal{I}}$ *is an interpretation function that maps every concept to a subset of* $\Delta^{\mathcal{I}}$ *and every role to a subset of* $(\Delta^{\mathcal{I}})^2$ *such that the following equations are satisfied.*

$$\begin{array}{rlrl} \top^{\mathcal{I}} &= \Delta^{\mathcal{I}} & (\exists S.C)^{\mathcal{I}} &= \{d \mid \exists d'.(d,d') \in S^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\}\\ \neg C^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} & (\leqslant k S.C)^{\mathcal{I}} &= \{d \mid \sharp\{d'.(d,d') \in S^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\} \leqslant k\}\\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} & R^{\mathcal{I}} &= (R^{\mathcal{I}})^+ \text{ for all } R \in \mathsf{NR}_+\\ & & (R^-)^{\mathcal{I}} &= \{(d',d) \mid (d,d') \in R^{\mathcal{I}}\} \end{array}$$

$\mathcal{SHIQ}$ schemas, ABoxes, and KBs are defined similarly to those for $\mathcal{DLR}$: if C, D are concepts, R, S are roles, and v, w are individuals, then a schema $\mathcal{S}$ consists of axioms of the form $C \sqsubseteq D$ and $R \sqsubseteq S$, and an ABox $\mathcal{A}$ consists of axioms of the form $w{:}C$ and $\langle v, w\rangle{:}R$. Again, a KB $\mathcal{K}$ is a pair $\langle \mathcal{S}, \mathcal{A}\rangle$, where $\mathcal{S}$ is a schema and $\mathcal{A}$ is an ABox.

The definitions of interpretations, satisfiability, and models also parallel those for $\mathcal{DLR}$, and there is again no unique name assumption.

Note that, in order to maintain decidability, the roles that can appear in number restrictions are restricted [21]: if a role S occurs in a number restriction $\leqslant kS.C$, then neither S nor any of its sub roles may be transitive (i.e., if the schema contains a $\sqsubseteq$-path from S' to S, then S' is not transitive).

3 Determining Query Containment

In this section we will describe how the problem of deciding whether one query is contained in another one w.r.t. a $\mathcal{DLR}$ schema can be reduced to the problem of deciding KB satisfiability in the $\mathcal{SHIQ}$ description logic. There are three steps to this reduction. Firstly, the queries are transformed into $\mathcal{DLR}$ ABoxes $\mathcal{A}_1$ and $\mathcal{A}_2$ such that $\mathcal{S} \models q_1 \sqsubseteq q_2$ iff $\langle \mathcal{S}, \mathcal{A}_1\rangle \approx\!\!\!|\, \mathcal{A}_2$ (see Definition 4). Secondly, the ABox inclusion problem is transformed into one or more KB satisfiability problems. Finally, we show how a $\mathcal{DLR}$ KB can be transformed into an equisatisfiable $\mathcal{SHIQ}$ KB.

3.1 Transforming Query Containment into ABox Inclusion

We will first show how a query can be transformed into a *canonical $\mathcal{DLR}$* ABox. Such an ABox represents a generic pattern that must be matched by all tuples in the evaluation of the query, similar to the tableau queries one encounters in the treatment of simple query containment for conjunctive queries [1].

Definition 6. *Let q be a conjunctive query. The* canonical ABox *for q is defined by*

$$\mathcal{A}_q = \{\boldsymbol{w}{:}\boldsymbol{R} \mid \boldsymbol{R}(\boldsymbol{w}) \text{ is an atom in } q\} \cup \{w{:}C \mid C(w) \text{ is an atom in } q\}.$$

We introduce a new atomic concept P_w for every individual w in $\mathcal{A}$ and define the completed *canonical ABox for q by*

$$\widehat{\mathcal{A}}_q = \mathcal{A}_q \cup \{w{:}P_w \mid w \text{ occurs in } \mathcal{A}_q\} \cup \{w_i{:}\neg P_{w_j} \mid w_i, w_j \text{ are constants in } q \text{ and } i \neq j\}.$$

The axioms $w{:}P_w$ in $\widehat{\mathcal{A}}_q$ introduce representative concepts *for each individual w in $\mathcal{A}_q$. They are used (in the axioms $w_i{:}\neg P_{w_j}$) to ensure that individuals corresponding to different constants in q cannot have the same interpretation, and will also be useful in the transformation to KB satisfiability.*

By abuse of notation, we will say that an interpretation $\mathcal{I}$ and an assignment ρ of distinguished variables, non-distinguished variables and constants to elements in the domain of $\mathcal{I}$ such that $\mathcal{I} \models \rho(q)$ define a model for $\mathcal{A}_q$ with the interpretation of the individuals corresponding with ρ and the interpretation $P_w^{\mathcal{I}} = \{w^{\mathcal{I}}\}$.

We can use this definition to transform the query containment problem into a (very similar) problem involving $\mathcal{DLR}$ ABoxes. We can assume that the names of the non-distinguished variables in q_2 differ from those in q_1 (arbitrary names can be chosen without affecting the evaluation of the query), and that the names of distinguished variables and constants appear in both queries (if a name is missing in one of the queries, it can be simply added using a term like $\top(v)$).

The following Theorem shows that a canonical ABox really captures the structure of a query, allowing the query containment problem to be restated as an ABox inclusion problem.

Theorem 1 *Given a schema $\mathcal{S}$ and two queries q_1 and q_2, $\mathcal{S} \models q_1 \sqsubseteq q_2$ iff $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle \models\!\!\approx \mathcal{A}_{q_2}$.*

Before we prove Theorem 1, note that, in general, this theorem no longer holds if we replace $\mathcal{A}_{q_2}$ by $\widehat{\mathcal{A}}_{q_2}$. Let $\mathcal{S}$ be a schema and q_1, q_2 be two queries such that q_1 is satisfiable w.r.t. $\mathcal{S}$ and q_2 contains at least one non-distinguished variable z. Then the completion $\widehat{\mathcal{A}}_{q_2}$ contains the assertion $z{:}P_z$ where P_z is a new atomic concept. Since q_1 is satisfiable w.r.t. $\mathcal{S}$ and P_z does not occur in $\mathcal{S}$ or q_1, $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle$ has a model $\mathcal{I}$ with $P_z^{\mathcal{I}} = \emptyset$. Such a model $\mathcal{I}$ cannot be extended to a model $\mathcal{I}'$ of $\widehat{\mathcal{A}}_{q_2}$ because there is no possible interpretation for z that would satisfy $z^{\mathcal{I}'} \in P_z^{\mathcal{I}'}$. Hence, $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle \not\models\!\!\approx \widehat{\mathcal{A}}_{q_2}$ regardless of whether $\mathcal{S} \models q_1 \sqsubseteq q_2$ holds or not. In the next section we will see how to deal with the non-distinguished individuals in $\mathcal{A}_{q_2}$ without the introduction of new representative concepts.

Proof. PROOF OF THEOREM 1: For the if direction, assume $\mathcal{S} \not\models q_1 \sqsubseteq q_2$. Then there exists a model $\mathcal{I}$ of $\mathcal{S}$ and a tuple $(d_1, \ldots, d_n) \in (\Delta^{\mathcal{I}})^n$ such that $(d_1, \ldots, d_n) \in q_1(\mathcal{I})$ and $(d_1, \ldots, d_n) \notin q_2(\mathcal{I})$. $\mathcal{I}$ and the assignment of variables leading to $(d_1, \ldots, d_n)$ define a model for $\widehat{\mathcal{A}}_{q_1}$. If $\cdot^{\mathcal{I}}$ could be extended to satisfy $\mathcal{A}_{q_2}$, then the extension would correspond to an assignment of the non-distinguished variables in q_2 such that $(d_1, \ldots, d_n) \in q_2(\mathcal{I})$, thus contradicting the assumption.

For the only if direction, assume there is a model $\mathcal{I}$ of both $\mathcal{S}$ and $\widehat{\mathcal{A}}_{q_1}$ that cannot be extended to a model of $\mathcal{A}_{q_2}$. Hence there is a tuple $(d_1, \ldots, d_n) \in q_1(\mathcal{I})$ and a corresponding assignment of variables that define $\mathcal{I}$. If there is an assignment of the non-distinguished variables in q_2 such that $(d_1, \ldots, d_n) \in q_2(\mathcal{I})$, then this assignment would define the extension of $\mathcal{I}$ such that $\mathcal{A}_{q_2}$ is also satisfied. $\square$

3.2 Transforming ABox Inclusion into ABox Satisfiability

Next, we will show how to transform the ABox inclusion problem into one or more KB satisfiability problems. In order to do this, there are two main difficulties that must be overcome. The first is that, in order to transform inclusion into satisfiability, we would like to be able to "negate" axioms. This is easy for axioms of the form $w{:}C$, because an interpretation satisfies $w{:}\neg C$ iff it does not satisfy $w{:}C$. However, we cannot deal with axioms of the form $\boldsymbol{w}{:}\mathbf{R}$ in this way, because $\mathcal{DLR}$ only has a weak form of negation for relations relative to $\top_n$. Our solution is to transform all axioms in $\mathcal{A}_{q_2}$ into the form $w{:}C$.

The second difficulty is that $\mathcal{A}_{q_2}$ may contain individuals corresponding to non-distinguished variables in q_2 (given the symmetry between queries and ABoxes, we will refer to them from now on as non-distinguished individuals). These individuals introduce an extra level of quantification that we cannot deal with using our standard reasoning procedures: $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle \approx\!\!\!\models \mathcal{A}_{q_2}$ iff for all models $\mathcal{I}$ of $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle$ *there exists* some extension of $\mathcal{I}$ to $\mathcal{A}_{q_2}$. We deal with this problem by eliminating the non-distinguished individuals from $\mathcal{A}_{q_2}$.

We will begin by exploiting some general properties of ABoxes that allow us to *compact* $\mathcal{A}_{q_2}$ so that it contains only one axiom $\boldsymbol{w}$:$\mathbf{R}$ for each tuple $\boldsymbol{w}$, and one axiom w:C for each individual w that is not an element in any tuple. It is obvious from the semantics that we can combine all ABox axioms relating to the same individual or tuple: $\mathcal{I} \models \{w{:}C, w{:}D\}$ (resp. $\{\boldsymbol{w}{:}\mathbf{R}, \boldsymbol{w}{:}\mathbf{S}\}$) iff $\mathcal{I} \models \{w{:}(C \sqcap D)\}$ (resp. $\{\boldsymbol{w}{:}(\mathbf{R} \sqcap \mathbf{S})\}$). The following lemma shows that we can also absorb $w_i{:}C$ into $\boldsymbol{w}{:}\mathbf{R}$ when w_i is an element of $\boldsymbol{w}$.

Lemma 1 *Let $\mathcal{A}$ be a $\mathcal{DLR}$ ABox with $\{w_i{:}C, \boldsymbol{w}{:}\boldsymbol{R}\} \subseteq \mathcal{A}$, where w_i is the ith element in $\boldsymbol{w}$. Then $\mathcal{I} \models \mathcal{A}$ iff $\mathcal{I} \models \{\boldsymbol{w}{:}(\boldsymbol{R} \sqcap \$i : C)\} \cup \mathcal{A} \setminus \{w_i{:}C, \boldsymbol{w}{:}\boldsymbol{R}\}$.*

PROOF: From the semantics, if $\boldsymbol{w}^{\mathcal{I}} \in (\mathbf{R} \sqcap \$i : C)^{\mathcal{I}}$, then $\boldsymbol{w}^{\mathcal{I}} \in \mathbf{R}^{\mathcal{I}}$ and $w_i^{\mathcal{I}} \in C^{\mathcal{I}}$, and if $w_i^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\boldsymbol{w}^{\mathcal{I}} \in \mathbf{R}^{\mathcal{I}}$, then $\boldsymbol{w}^{\mathcal{I}} \in (\mathbf{R} \sqcap \$i : C)^{\mathcal{I}}$. $\square$

The ABox resulting from exhaustive application of Lemma 1 can be represented as a graph, with a node for each tuple, a node for each individual, and edges connecting tuples with the individuals that compose them. The graph will consist of one or more connected components, where each component is either a single individual (representing an axiom $w{:}C$, where w is not an element in any tuple) or a set of tuples linked by common elements (representing axioms of the form $\boldsymbol{w}{:}\mathbf{R}$). As the connected components do not have any individuals in common, we can deal independently with the inclusion problem for each connected set of axioms: $\langle \mathcal{S}, \mathcal{A} \rangle \approx\!\!\!\models \mathcal{A}'$ iff $\langle \mathcal{S}, \mathcal{A} \rangle \approx\!\!\!\models \mathcal{G}$ for every connected set of axioms $\mathcal{G} \subseteq \mathcal{A}'$. As an example, Figure 1 shows the graph that corresponds to the ABox $\mathcal{A}_{q_2}$ from Example 1.

Returning to our original problem, we will now show how we can *collapse* a connected component $\mathcal{G}$ by a graph traversal into a single axiom of the form $w{:}C$, where w is an element of a tuple occurring in $\mathcal{G}$ (an arbitrarily chosen “root” individual), and C is a concept that describes $\mathcal{G}$ from the point of view of w. An example for this process will be given later in this section.

This would be easy if we were able to refer to individuals in C (i.e., if our logic included *nominals* [25]), which is not the case. However, as we will see, it is sufficient to refer to the distinguished individuals w_i in $\mathcal{G}$ (which also occur in $\widehat{\mathcal{A}}_{q_1}$) by their representative concepts P_{w_i}. Moreover, we can refer to non-distinguished individuals z_i by using $\top$ as their representative concept (this is only valid for z_i that are encountered only once during the traversal of $\mathcal{G}$, but we will see later that we can, without loss of generality, restrict our attention to this case). Informally, the use of $\top$ as the representative concept for such z_i can be justified by the fact that when an interpretation $\mathcal{I}$ is extended to $\mathcal{G}$, z_i can be interpreted as any element in $\Delta^{\mathcal{I}}$ $(= \top^{\mathcal{I}})$.[3]

[3] For full details, the reader is again referred to [20].

The following lemma shows how we can use the representative concepts to transform an axiom of the form $\boldsymbol{w}{:}\mathbf{R}$ into an axiom of the form $w_i{:}C$.

Lemma 2 *If $\mathcal{S}$ is a schema, $\widehat{\mathcal{A}}$ is a completed canonical ABox and $\mathcal{A}'$ is an ABox with $\boldsymbol{w}{:}\boldsymbol{R} \in \mathcal{A}'$, then $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle \approx \mathcal{A}'$ iff $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle \approx (\{w_i{:}C\} \cup \mathcal{A}' \setminus \{\boldsymbol{w}{:}\boldsymbol{R}\})$, where $\boldsymbol{w} = \langle w_1, \dots, w_n \rangle$, w_i is the ith element in $\boldsymbol{w}$, C is the concept*

$$\exists[\$i](\boldsymbol{R} \sqcap \bigsqcap_{1 \leqslant j \leqslant n . j \neq i} (\$j/n : P_j)),$$

and P_j is the appropriate representative concept for w_j ($\top$ if w_j is a non-distinguished individual, P_{w_j} otherwise).

PROOF (sketch): For the only if direction, it is easy to see that, if $\mathcal{I} \models \langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle$, and $\mathcal{I}'$ is an extension of $\mathcal{I}$ that satisfies $\boldsymbol{w}{:}\mathbf{R}$, then $\mathcal{I}'$ also satisfies $w_i{:}C$.

The converse direction is more complicated, and exploits the fact that, for every model $\mathcal{I}$ of $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle$, there is a similar model $\mathcal{I}'$ in which every representative concept P_{w_i} is interpreted as $\{w_i^{\mathcal{I}'}\}$. If $\mathcal{I}$ cannot be extended to satisfy $\boldsymbol{w}{:}\mathbf{R}$, then neither can $\mathcal{I}'$, and, given the interpretations of the P_{w_i}, it is possible to show that $\mathcal{I}'$ cannot be extended to satisfy $w_i{:}C$ either. $\square$

All that now remains is to choose the order in which we apply the transformations from Lemma 1 and 2 to the axioms in $\mathcal{G}$, so that, whenever we use Lemma 2 to transform $\boldsymbol{w}{:}\mathbf{R}$ into $w_i{:}C$, we can then use Lemma 1 to absorb $w_i{:}C$ into another axiom $\boldsymbol{v}{:}\mathbf{R}$, where w_i is an element of $\boldsymbol{v}$. We can do this using a recursive traversal of the graphical representation of $\mathcal{G}$ (a similar technique is used in [4] to transform queries into concepts). A traversal starts at an individual node w (the "root") and proceeds as follows.

– At an individual node w_i, the node is first marked as visited. Then, while there remains an unmarked tuple node connected to w_i, one of these, $\boldsymbol{w}$, is selected, visited, and the axiom $\boldsymbol{w}{:}\mathbf{R}$ transformed into an axiom $w_i{:}C$. Finally, any axioms $w_i{:}C_1, \dots, w_i{:}C_n$ resulting from these transformations are merged into a single axiom $w_i{:}(C_1 \sqcap \dots \sqcap C_n)$.
– At a tuple node $\boldsymbol{w}$, the node is first marked as visited. Then, while there remains an unmarked individual node connected to $\boldsymbol{w}$, one of these, w_i, is selected, visited, and any axiom $w_i{:}C$ that results from the visit is merged into the axiom $\boldsymbol{w}{:}\mathbf{R}$ using Lemma 1.

Note that the correctness of the collapsing procedure does not depend on the traversal (whose purpose is simply to choose a suitable ordering), but only on the individual transformations.

Having collapsed a component $\mathcal{G}$, we finally have a problem that we can decide using KB satisfiability:

Lemma 3 *If $\mathcal{S}$ is a schema and $\widehat{\mathcal{A}}$ is a completed canonical ABox, then $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle \approx \{w{:}C\}$ iff w is an individual in $\widehat{\mathcal{A}}$ and $\langle \mathcal{S}, (\widehat{\mathcal{A}} \cup \{w{:}\neg C\}) \rangle$ is not satisfiable, or w is not an individual in $\widehat{\mathcal{A}}$ and $\langle (\mathcal{S} \cup \{\top \sqsubseteq \neg C\}), \widehat{\mathcal{A}} \rangle$ is not satisfiable.*

PROOF (sketch): If w is an individual in $\widehat{\mathcal{A}}$, $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle \approx \{w{:}C\}$ implies that every model $\mathcal{I}$ of $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle$ must also satisfy $w{:}C$, and this is true iff $\mathcal{I}$ does not satisfy $w{:}\neg C$. In the case where w is not an individual in $\widehat{\mathcal{A}}$, a model $\mathcal{I}$ of $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle$ can be extended to $\{w{:}C\}$ iff $C^{\mathcal{I}} \neq \emptyset$, which is true iff $\Delta^{\mathcal{I}} \nsubseteq (\neg C)^{\mathcal{I}}$. $\square$

If a non-distinguished individual z_i is encountered more than once during a traversal, then it is enforcing a co-reference that closes a cycle in the query. In this case we cannot simply use $\top$ to refer to it, as this would fail to capture the fact that z_i must be interpreted as the same element of $\Delta^{\mathcal{I}}$ on each occasion.

In [4] this problem is dealt with by replacing the non-distinguished variables occurring in a cycle in q_2 with variables or constants from q_1, and forming a disjunction of the concepts resulting from each possible replacement. This is justified by the fact that cycles cannot be expressed in the $\mathcal{DLR}$ schema and so must be present in q_1. However, this fails to take into account the fact that identifying two or more of the non-distinguished variables in q_2 could eliminate the cycle.

We overcome this problem by introducing an additional layer of disjunction in which non-distinguished individuals occurring in cycles are identified (in every possible way) with other individuals occurring in the same cycle. We then continue as in [4], but only replacing those individuals that actually enforce a co-reference, i.e., that would be encountered more than once during the graph traversal.[4]

Example 1 To illustrate the inclusion to satisfiability transformation, we will refer to the example given in Section 2.2. The containment of BUS in CITY_BUS w.r.t. the schema is demonstrated by the inclusion $\langle \mathcal{S}, \widehat{\mathcal{A}}_1 \rangle \approx \mathcal{A}_2$, where $\mathcal{S}$, $\widehat{\mathcal{A}}_1$ and $\mathcal{A}_2$ are the schema and two canonical ABoxes (completed in the case of $\widehat{\mathcal{A}}_1$) corresponding to the given queries:

$$\begin{aligned}
\mathcal{S} &= \left\{ \begin{array}{l} (\text{bus\_route} \sqcap (\$1/3 : \text{city\_bus})) \sqsubseteq \text{city\_bus\_route}, \\ \text{city\_bus\_route} \sqsubseteq (\text{bus\_route} \sqcap (\$1/3 : \text{city\_bus})) \end{array} \right\} \\
\widehat{\mathcal{A}}_1 &= \left\{ \langle n, y_1, y_2 \rangle{:}\text{bus\_route}, \langle n, y_2, y_1 \rangle{:}\text{bus\_route}, n{:}\text{city\_bus}, n{:}P_n, y_1{:}P_{y_1}, y_2{:}P_{y_2} \right\} \\
\mathcal{A}_2 &= \left\{ \langle n, z_1, z_2 \rangle{:}\text{city\_bus\_route}, \langle n, z_2, z_1 \rangle{:}\text{city\_bus\_route} \right\}
\end{aligned}$$

The two axioms in $\mathcal{A}_2$ are connected, and can be collapsed into a single axiom using the described procedure. Figure 1 shows a traversal of the graph $\mathcal{G}$ corresponding to $\mathcal{A}_{q_2}$ that starts at z_1 and traverses the edges in the indicated sequence.[5] The resulting axiom (describing $\mathcal{A}_2$ from the point of view of z_1) is $z_1{:}C$, where C is the concept

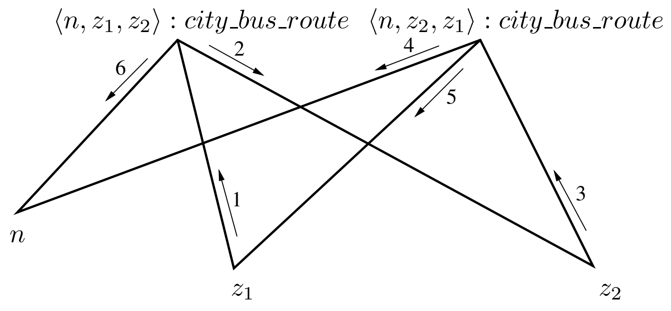
$$\underset{1}{\exists}[\$2](\text{city\_bus\_route} \sqcap (\underset{2}{\$3} : (P_{z_2} \sqcap \underset{3}{\exists}[\$2](\text{city\_bus\_route} \sqcap \underset{4}{\$1} : P_n \sqcap \underset{5}{\$3} : P_{z_1}))) \sqcap \underset{6}{\$1} : P_n)$$

P_{z_1}, P_{z_2} are "place-holders" for z_1, z_2[6] and the numbers below the $\mathcal{DLR}$ operators denote the edges which correspond to the respective subconcept of C. As z_2 is encountered only once in the traversal, P_{z_2} can be replaced with $\top$, but as z_1 is encountered

[4] Note that the graph traversal must always start from the same root.

[5] We will ignore the first non-deterministic step as no individual identifications are required in order to prove the containment.

[6] In practice, we use such "place-holders" during the collapsing procedure and then make appropriate (possibly non-deterministic) substitutions.



$\langle n, z_1, z_2\rangle : city\_bus\_route$
$\langle n, z_2, z_1\rangle : city\_bus\_route$
6
2
4
5
1
3
n
z_1
z_2

Fig. 1. A traversal of the graph corresponding to $\mathcal{A}_{q_2}$

twice (as the root and as P_{z_1}), it must be replaced (non-deterministically) with an individual i occurring in $\widehat{\mathcal{A}}_1$ (we will refer to the resulting concepts as $C_{[z_1/i]}$), and thus $\langle \mathcal{S}, \widehat{\mathcal{A}}_1\rangle \models\!\!\approx \mathcal{A}_2$ iff $\langle \mathcal{S}, \widehat{\mathcal{A}}_1\rangle \models\!\!\approx \{i{:}C_{[z_1/i]}\}$. Taking $i = y_1$ we have $\langle \mathcal{S}, \widehat{\mathcal{A}}_1\rangle \models\!\!\approx \{y_1{:}C_{[z_1/y_1]}\}$ because $\langle \mathcal{S}, (\widehat{\mathcal{A}}_1 \cup \{y_1{:}\neg C_{[z_1/y_1]}\})\rangle$ is not satisfiable.

Summing up, we thus have:

Theorem 2 *For a $\mathcal{DLR}$ KB $\mathcal{K} = \langle \mathcal{S}, \mathcal{A}\rangle$ and a $\mathcal{DLR}$ ABox $\mathcal{A}'$, the problem of deciding whether $\mathcal{A}$ is included in $\mathcal{A}'$ w.r.t. $\mathcal{S}$ can be reduced to (possibly several) $\mathcal{DLR}$ ABox satisfiability problems.*

Concerning the practicability of this reduction, it is easy to see that, for any fixed choice of substitutions for the non-distinguished individuals in $\mathcal{G}$, the reduction from Theorem 2 can be computed in polynomial time. More problematically, it is necessary to consider each possible identification of non-distinguished individuals occuring in cycles in $\mathcal{G}$, and for each of these *all* possible mappings from the set Z of non-distinguished individuals that occur more than once in the collapsed $\mathcal{G}$ to to the set W of individuals that occur in $\widehat{\mathcal{A}}_1$ (of which there are $|W|^{|Z|}$ many). However, both Z and W will typically be quite small, especially Z which will consist only of those non-distinguished individuals that occur in a cycle in $\mathcal{G}$ and are actually used to enforce a co-reference (i.e., to “close” the cycle). This represents a useful refinement over the procedure described in [4], where all z_i that occur in cycles are non-deterministically replaced with some w_i, regardless of whether or not they are used to enforce a co-reference. Moreover, it is easy to show that most individual identifications cannot contribute to the solution, and can thus be ignored. Therefore, we do not believe that this additional non-determinism compromises the feasibility of our approach.

Interestingly, also in [13], cycles in queries are identified as a main cause for complexity. There it is shown that query containment without constraints is decidable in polynomial time for acyclic queries whereas the problem for possibly cyclic queries is NP-complete [12].

3.3 Transforming $\mathcal{DLR}$ satisfiability into $\mathcal{SHIQ}$ satisfiability

We decide satisfiability of $\mathcal{DLR}$ knowledge bases by means of a satisfiability-preserving translation $\sigma(\cdot)$ from $\mathcal{DLR}$ KBs to $\mathcal{SHIQ}$ KBs. This translation must deal with the fact that $\mathcal{DLR}$ allows for arbitrary n-ary relations while $\mathcal{SHIQ}$ only allows for unary predicates and binary relations; this is achieved by a process called *reification* (see, for example [16]). The main idea behind this is easily described: each n-ary tuple in a $\mathcal{DLR}$-interpretation is represented by an individual in a $\mathcal{SHIQ}$-interpretation that is linked via the dedicated functional relations $f_1, \ldots, f_n$ to the elements of the tuple.

For $\mathcal{DLR}$ without regular expressions, the mapping $\sigma(\cdot)$ (given by [4])

$$\begin{array}{rl@{\quad}rl}
\sigma(\top_n) &= \top_n & \sigma(\top) &= \top_1 \\
\sigma(\mathbf{P}) &= \mathbf{P} & \sigma(A) &= A \\
\sigma(\$i/n : C) &= \top_n \sqcap \exists f_i.\sigma(C) & \sigma(\neg C) &= \neg\sigma(C) \\
\sigma(\neg\mathbf{R}) &= \top_n \sqcap \neg\sigma(\mathbf{R}) & \sigma(C_1 \sqcap C_2) &= \sigma(C_1) \sqcap \sigma(C_2) \\
\sigma(\mathbf{R}_1 \sqcap \mathbf{R}_2) &= \sigma(\mathbf{R}_1) \sqcap \sigma(\mathbf{R}_2) & \sigma(\exists[\$i]\mathbf{R}) &= \exists f_i^-.\sigma(\mathbf{R}) \\
 & & \sigma(\leq k[\$i]\mathbf{R}) &= (\leq k\ f_i^-\ \sigma(\mathbf{R}))
\end{array}$$

reifies $\mathcal{DLR}$ expressions into $\mathcal{SHIQ}$-concepts. This mapping can be extended to a knowledge base (KB) as follows.

Definition 7. *Let $\mathcal{K} = (\mathcal{S}, \mathcal{A})$ be a $\mathcal{DLR}$ KB. The reification of $\mathcal{S}$ is given by*

$$\{(\sigma(\boldsymbol{R}_1) \sqsubseteq \sigma(\boldsymbol{R}_2)) \mid (\boldsymbol{R}_1 \sqsubseteq \boldsymbol{R}_2) \in \mathcal{S}\} \cup \{(\sigma(C_1) \sqsubseteq \sigma(C_2)) \mid (C_1 \sqsubseteq C_2) \in \mathcal{S}\}.$$

To reify the ABox $\mathcal{A}$, we have to reify all tuples appearing in the axioms. For each distinct tuple $\boldsymbol{w} = \langle w_1, \ldots, w_n\rangle$ occurring in $\mathcal{A}$, we chose a distinct individual $t_{\boldsymbol{w}}$ (called the "reification of $\boldsymbol{w}$") and define:

$$\sigma(\boldsymbol{w}{:}\boldsymbol{R}) = \{t_{\boldsymbol{w}}{:}\sigma(\boldsymbol{R})\} \cup \{\langle t_{\boldsymbol{w}}, w_i\rangle{:}f_i \mid 1 \leq i \leq n\} \quad \textit{and}$$
$$\sigma(\mathcal{A}) = \bigcup \{\sigma(\boldsymbol{w}{:}\boldsymbol{R}) \mid \boldsymbol{w}{:}\boldsymbol{R} \in \mathcal{A}\} \cup \ \{w{:}\sigma(C) \mid w{:}C \in \mathcal{A}\}.$$

We need a few additional inclusion and ABox axioms to guarantee that any model of $(\sigma(\mathcal{S}), \sigma(\mathcal{A}))$ can be "un-reified" into a model of $(\mathcal{S}, \mathcal{A})$. Let n_{max} denote the maximum arity of the $\mathcal{DLR}$ relations appearing in $\mathcal{K}$. We define $f(\mathcal{S})$ to consist of the following axioms (where $x \equiv y$ is an abbreviation for $x \sqsubseteq y$ and $y \sqsubseteq x$):

$$\begin{array}{rl@{\qquad}l}
\top &\equiv \top_1 \sqcup \cdots \sqcup \top_{n_{max}} & \\
\top &\sqsubseteq (\leq 1\ f_1) \sqcap \cdots \sqcap (\leq 1\ f_{n_{max}}) & \\
\forall f_i.\bot &\sqsubseteq \forall f_{i+1}.\bot & \textit{for } 2 \leq i < n_{max} \\
\top_i &\equiv \exists f_1.\top_1 \sqcap \cdots \sqcap \exists f_i.\top_1 \sqcap \forall f_{i+1}.\bot & \textit{for } 2 \leq i \leq n_{max} \\
\boldsymbol{P} &\sqsubseteq \top_n & \textit{for each atomic relation } \boldsymbol{P} \textit{ of arity } n \\
A &\sqsubseteq \top_1 & \textit{for each atomic concept } A
\end{array}$$

These are standard reification axioms, and can already be found in [4].

We introduce a new atomic concept Q_w for every individual w in $\mathcal{A}$ and define $f(\mathcal{A})$ to consist of the following axioms:

$$\begin{aligned}
f(\mathcal{A}) = \{&w{:}Q_w \mid w \textit{ occurs in } \mathcal{A}\} \cup \\
&\{w_1{:}\leqslant 1\, f_1^-.(\top_n \sqcap \exists f_2.Q_{w_2} \sqcap \ldots \sqcap \exists f_n.Q_{w_n}) \mid \langle w_1, \ldots, w_n\rangle \textit{ occurs in } \mathcal{A}\}
\end{aligned}$$

These axioms are crucial when dealing with the problem of tuple-admissibility (see below) in the presence of ABoxes.

Finally, we define $\sigma(\mathcal{K}) = \langle(\sigma(\mathcal{S}) \cup f(\mathcal{S})), (\sigma(\mathcal{A}) \cup f(\mathcal{A}))\rangle$.

Theorem 3 *Let $\mathcal{K} = \langle\mathcal{S}, \mathcal{A}\rangle$ be a $\mathcal{DLR}$ knowledge-base. $\mathcal{K}$ is satisfiable iff the $\mathcal{SHIQ}$-KB $\sigma(\mathcal{K})$ is satisfiable.*

PROOF (sketch): The same techniques that were used in [2] can be adapted to the DL $\mathcal{SHIQ}$, and extended to deal with ABox axioms. The only-if direction is straightforward. A model $\mathcal{I}$ of $\mathcal{K}$ can be transformed into a model of $\sigma(\mathcal{K})$ by introducing, for every arity n with $2 \leq n \leq n_{\text{max}}$ and every n-tuple of elements $\boldsymbol{d} \in (\Delta^{\mathcal{I}})^n$, a new element $t_{\boldsymbol{d}}$ that is linked to the elements of $\boldsymbol{d}$ by the functional relations $f_1, \ldots, f_n$. If we interpret $\top_1$ by $\Delta^{\mathcal{I}}$, $\top_n$ by the reifications of all elements in $\top_n^{\mathcal{I}}$, and, for every w that occurs in $\mathcal{A}$, Q_w by $w^{\mathcal{I}}$, then it is easy to show that we have constructed a model of $\sigma(\mathcal{K})$.

The converse direction is more complicated since a model of $\sigma(\mathcal{K})$ is not necessarily *tuple-admissible*, i.e., in general there may be distinct elements t, t' that are reifications of the same tuple $\boldsymbol{d}$. In the “un-reification” of such a model, $\boldsymbol{d}$ would only appear once which may conflict with assertions in the $\mathcal{DLR}$ KB about the number of tuples in certain relations. However, it can be shown that every satisfiable KB $\sigma(\mathcal{K})$ also has a tuple-admissible model. It is easy to show that such a model, by “un-reification”, induces a model for the original KB $\mathcal{K}$. $\square$

We now have the machinery to transform a query containment problem into one or more $\mathcal{SHIQ}$ schema and ABox satisfiability problems. In the FaCT system we already have a decision procedure for $\mathcal{SHIQ}$ schema satisfiability, and this can be straightforwardly extended to deal with ABox axioms [22].

We have already argued why we believe our approach to be feasible. It should also be mentioned, that our approach matches the known worst-case complexity of the problem, which was determined as EXPTIME-complete in [4]. Satisfiability of a $\mathcal{SHIQ}$-KB can be determined in EXPTIME.[7] All reduction steps can be computed in deterministic polynomial time, with the exception of the reduction used in Theorem 2, which requires consideration of exponentially many mappings. Yet, for every fixed mapping, the reduction is polynomial, which yields that our approach decides query containment in EXPTIME.

4 The FaCT System

It is claimed in Section 1 that one of the main benefits of our approach is that it leads to a practical solution to the query containment problem. In this section we will substantiate this claim by presenting the results of a simple experiment in which the FaCT system is used to decide a query containment problem with respect to the DWQ schema mentioned in Section 1.

[7] This does not follow from the algorithm presented in [22], which focuses on feasibility rather than worst-case complexity. It can be shown using a precompletion strategy similar to the one used in [26] together with the EXPTIME-completeness of $\mathcal{CIQ}$ [15].

The FaCT system includes an optimised implementation of a schema satisfiability testing algorithm for the DL $\mathcal{SHIQ}$. As the extension of FaCT to include the ABox satisfiability testing algorithm described in [22] has not yet been completed, FaCT is currently only able to test the satisfiability of a KB $\langle \mathcal{S}, \mathcal{A} \rangle$ in the case where the $\mathcal{A}$ contains a single axiom of the form $w{:}C$ (this is equivalent to testing the satisfiability of the concept C w.r.t. the schema $\mathcal{S}$). We have therefore chosen a query containment problem that can be reduced to a $\mathcal{SHIQ}$ KB satisfiability problem of this form using the methodology described in Section 3.

The DWQ schema is derived from the integration of several Extended Entity-Relationship (EER) schemas using $\mathcal{DLR}$ axioms to define inter-schema constraints [7]. One of the schemas, called the *enterprise* schema, represents the global concepts and relationships that are of interest in the Data Warehouse; a fragment of the enterprise schema that will be relevant to the query containment example is shown in Figure 2. A total of 5 source schemas representing (portions of) actual data sources are integrated with the enterprise schema using $\mathcal{DLR}$ axioms to establish the relationship between entities and relations in the source and enterprise schemas (the resulting integrated schema contains 48 entities, 29 relations and 49 $\mathcal{DLR}$ axioms). For example, one of the $\mathcal{DLR}$ axioms defining the relationship between the enterprise schema and the entity "Business-Customer" in the source schema describing business contracts is

$$\begin{aligned}\text{Business-Customer} \sqsubseteq (&\text{Company} \sqcap \exists[\$1](\text{agreement} \sqcap \\ &(\$2/3 : (\text{Contract} \sqcap \exists[\$1](\text{contract-company} \sqcap \\ &\quad(\$2/2 : \text{Telecom-company})))))).\end{aligned}$$

This axiom states, roughly speaking, that a Business-Customer is a kind of Company that has an agreement where the contract is with a Telecom-company.

As a result of this axiom, it is relatively easy to see that the query

$$q_1(x) \leftarrow \text{Business-Customer}(x)$$

is contained in the query

$$\begin{aligned}q_2(x) \leftarrow\ &\text{agreement}(x, y_1, y_2) \wedge \text{Contract}(y_1) \wedge \text{Service}(y_2) \wedge \\ &\text{contract-company}(y_1, y_3) \wedge \text{Telecom-company}(y_3)\end{aligned}$$

with respect to the DWQ schema $\mathcal{S}$, written $\mathcal{S} \models q_1 \sqsubseteq q_2$.

The two queries can be transformed into the following (completed) canonical $\mathcal{DLR}$ ABoxes

$$\begin{aligned}\widehat{\mathcal{A}}_{q_1} &= \{x{:}\text{Business-Customer}, x{:}P_x\} \\ \mathcal{A}_{q_2} &= \{\langle x, y_1, y_2\rangle{:}\text{agreement}, y_1{:}\text{Contract}, y_2{:}\text{Service}, \\ &\qquad \langle y_1, y_3\rangle{:}\text{contract-company}, y_3{:}\text{Telecom-company}\},\end{aligned}$$

where P_x is the representative concept for x. We can now compact and collapse $\mathcal{A}_{q_2}$ to give an ABox $\{x{:}C_{q_2}\}$, where

$$\begin{aligned}C_{q_2} = \exists[\$1](&\text{agreement} \sqcap (\$2/3 : P_{y_1}) \sqcap (\$3/3 : P_{y_2}) \sqcap (\$2/3 : \text{Contract}) \sqcap \\ &(\$3/3 : \text{Service}) \sqcap (\$2/3 : (\exists[\$1]\, \text{contract-company} \sqcap (\$2/2 : P_{y_3}) \sqcap \\ &\qquad (\$2/2 : \text{Telecom-company})))).\end{aligned}$$

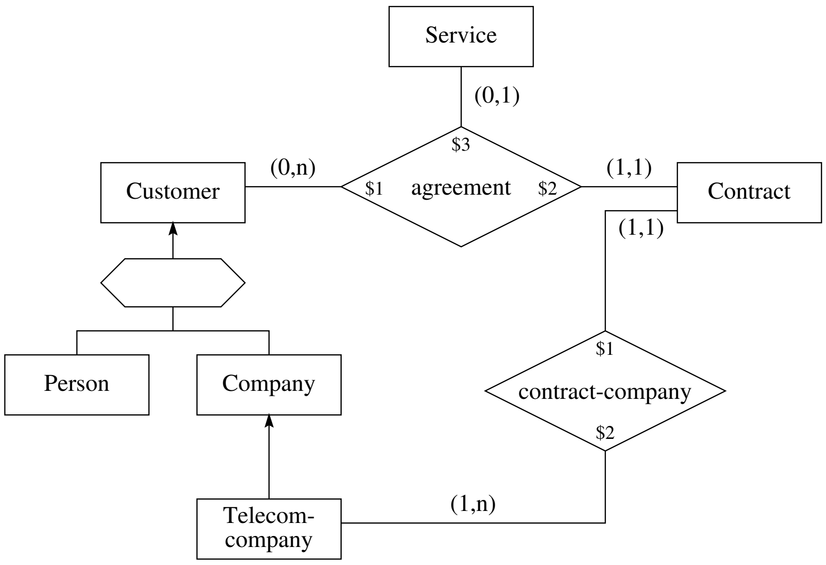


Fig. 2. A fragment of the DWQ enterprise schema

As each of the place-holders P_{y_1}, P_{y_2} and P_{y_3} occurs only once in the ABox, they can be replaced with $\top$, and C_{q_2} can be simplified to give

$$\begin{aligned} C'_{q_2} = {} & \exists[\$1](\text{agreement} \sqcap (\$2/3 : \text{Contract}) \sqcap (\$3/3 : \text{Service}) \sqcap \\ & (\$2/3 : (\exists[\$1]\text{contract-company} \sqcap (\$2/2 : \text{Telecom-company})))). \end{aligned}$$

We can now determine if the query containment $\mathcal{S} \models q_1 \sqsubseteq q_2$ holds by testing the satisfiability of the KB $\langle \mathcal{S}, \mathcal{A} \rangle$, where $\mathcal{A} = \{x\text{:Business-Customer}, x{:}P_x, x{:}\neg C'_{q_2}\}$. Moreover, $\mathcal{A}$ can be compacted to give $\{x{:}C\}$, where $C = \text{Business-Customer} \sqcap P_x \sqcap \neg C'_{q_2}$, and the KB satisfiability problem can be decided by using FaCT to test the satisfiability of the concept $\sigma(C)$ w.r.t. the schema $\sigma(\mathcal{S})$. Thus we have $\mathcal{S} \models q_1 \sqsubseteq q_2$ iff $\sigma(C)$ is not satisfiable w.r.t. $\sigma(\mathcal{S})$.

The FaCT system is implemented in Common Lisp, and the tests were performed using Allegro CL Enterprise Edition 5.0 running under Red Hat Linux on a 450MHz Pentium III with 128Mb of RAM. Excluding the time taken to load the schema from disk (60ms), FaCT takes only 60ms to determine that $\sigma(C)$ is not satisfiable w.r.t. $\sigma(\mathcal{S})$. Moreover, if $\sigma(\mathcal{S})$ is first *classified* (i.e., the subsumption partial ordering of all named concepts in $\sigma(\mathcal{S})$ is computed and cached), the time taken to determine the unsatisfiability is reduced to only 20ms. The classification procedure itself takes 3.5s (312 satisfiability tests are performed at an average of $\approx$11ms per satisfiability test), but this only needs to be done once for a given schema.

Although the above example is relatively trivial, it still requires FaCT to perform quite complex reasoning, the result of which depends on the presence of $\mathcal{DLR}$ inter-schema constraint axioms; in the absence of such axioms (e.g., in the case of a single

EER schema), reasoning should be even more efficient. Of course deciding arbitrary query containment problems would, in general, require full ABox reasoning. However, the above tests still give a useful indication of the kind of performance that could be expected: the algorithm for deciding $\mathcal{SHIQ}$ ABox satisfiability presented [22] is similar to the algorithm implemented in FaCT, and as the number of individuals generated by the encoding of realistic query containment problems will be relatively small, extending FaCT to deal with such problems should not compromise the demonstrated empirical tractability. Moreover, given the kind of performance exhibited by FaCT, the limited amount of additional non-determinism that might be introduced as a result of cycles in the containing query would easily be manageable.

The results presented here are also substantiate our claim that transforming $\mathcal{DLR}$ satisfiability problems into $\mathcal{SHIQ}$ leads to greatly improved empirical tractability with respect to the embedding technique described in Calvanese et al. [4]. During the DWQ project, attempts were made to classify the DWQ schema using a similar embedding in the less expressive $\mathcal{SHIF}$ logic [19] implemented in an earlier version of the FaCT system. These attempts were abandoned after several days of CPU time had been spent in an unsuccessful effort to solve a single satisfiability problem. This is in contrast to the 3.5s taken by the new $\mathcal{SHIQ}$ reasoner to perform the 312 satisfiability tests required to classify the whole schema.

5 Discussion

In this paper we have shown how the problem of query containment under constraints can be decided using a KB (schema plus ABox) satisfiability tester for the $\mathcal{SHIQ}$ description logic, and we have indicated how a $\mathcal{SHIQ}$ schema satisfiability testing algorithm can be extended to deal with an ABox. We have only talked about conjunctive queries, but extending the procedure to deal with disjunctions of conjunctive queries is straightforward. The procedure for verifying containment between disjunctions of conjunctive queries is not very different from the one described for simple conjunctive queries. The main difference is that, although each conjunctive part becomes an ABox (as described in Section 3.1), the object representing the whole disjunctive query is a set of alternative ABoxes. This results in one more non-deterministic step, whose complexity is determined by the number of disjuncts appearing in both queries. Full details can be found in [20].

Although there is some loss of expressive power with respect to the framework presented in [4] this seems to be acceptable when modelling classical relational information systems, where regular expressions are seldom used.

As we have shown in Section 4, the FaCT implementation of the $\mathcal{SHIQ}$ schema satisfiability algorithm works well with realistic problems, and given that the number of individuals generated by query containment problems will be relatively small, there is good reason to believe that a combination of the ABox encoding and the extended algorithm will lead to a practical decision procedure for query containment problems. Work is underway to test this hypothesis by extending the FaCT system to deal with $\mathcal{SHIQ}$ ABoxes.

Bibliography

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases*. Addison-Wesley, 1995.

[2] D. Calvanese. *Unrestricted and Finite Model Reasoning in Class-Based Representation Formalisms*. PhD thesis, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1996.

[3] D. Calvanese, G. De Giacomo, and M. Lenzerini. Structured objects: modeling and reasoning. In *Proc. of DOOD'95*, number 1013 in LNCS, pages 229–246. Springer-Verlag, 1995.

[4] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS'98*, 1998.

[5] D. Calvanese, G. De Giacomo, and M. Lenzerini. Answering queries using views in description logics. In *Proc. of KRDB'96*, pages 6–10. CEUR, 1999.

[6] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Source integration in data warehousing. In *Proc. of DEXA-98*, pages 192–197. IEEE Computer Society Press, 1998.

[7] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Use of the data reconciliation tool at telecom italia. DWQ deliverable D.4.3, Foundations of Data Warehouse Quality (DWQ), 1999.

[8] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In Jan Chomicki and Günter Saake, editors, *Logics for Databases and Information Systems*, pages 229–263. Kluwer Academic Publisher, 1998.

[9] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. Description logic framework for information integration. In *Proc. of KR-98*, pages 2–13, San Francisco, 1998. Morgan Kaufmann.

[10] T. Catarci and M. Lenzerini. Representing and using interschema knowledge in cooperative information systems. *Journal for Intelligent and Cooperative Information Systems*, 2(4):375–399, 1993.

[11] Edward P. F. Chan. Containment and minimization of positive conjunctive queries in OODB's. In *Proc. of PODS'92*, pages 202–211. ACM Press, 1992.

[12] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proc. of 9th Annual ACM Symposium on the Theory of Computing*, pages 77–90. Assoc. for Computing Machinery, 1977.

[13] C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. In *Proc. of ICDT-97*, number 1186 in LNCS, pages 56–70. Springer-Verlag, 1997.

[14] G. De Giacomo. *Decidability of Class-Based Knowledge Representation Formalisms*. PhD thesis, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1995.

[15] G. De Giacomo and M. Lenzerini. TBox and ABox reasoning in expressive description logics. In *Proc. of KR'96*, pages 316–327. Morgan Kaufmann, 1996.

[16] Giuseppe De Giacomo and Maurizio Lenzerini. What's in an aggregate: foundations for description logics with tuples and sets. In *Proc. of IJCAI'95*, pages 801–807. Morgan Kaufmann, 1995.

[17] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.

[18] I. Horrocks. FaCT and iFaCT. In *Proc. of DL'99*, pages 133–135. CEUR, 1999.

[19] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410, 1999.

[20] I. Horrocks, U. Sattler, S. Tessaris, and S. Tobies. Query containment using a DLR ABox. LTCS-Report 99-15, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1999. See http://www-lti.informatik.rwth-aachen.de/Forschung/Reports.html.

[21] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. of LPAR'99*, pages 161–180, 1999.

[22] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic shiq. In *Proc. of CADE-17*, number 1831 in LNCS, pages 482–496. Springer-Verlag, 2000.

[23] D. S. Johnson and A. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *Journal of Computer and System Sciences*, 28(1):167–189, 1984.

[24] A. Y. Levy and M.-C. Rousset. CARIN: A representation language combining horn rules and description logics. In *Proc. of ECAI'96*, pages 323–327. John Wiley, 1996.

[25] A. Schaerf. Reasoning with individuals in concept languages. *Data and Knowledge Engineering*, 13(2):141–176, 1994.

[26] S. Tessaris and G. Gough. Abox reasoning with transitive roles and axioms. In *Proc. of DL'99*. CEUR, 1999.