How to decide Query Containment under Constraints using a Description Logic

Ian Horrocks and Sergio Tessaris
Department of Computer Science
University of Manchester, UK
<*lastname*>@cs.man.ac.uk

Ulrike Sattler and Stephan Tobies
LuFg Theoretical Computer Science
RWTH Aachen, Germany
<*lastname*>@informatik.rwth-aachen.de

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 [7]) 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 [4]).

Query containment have, e.g., been studied in [9, 19, 8, 20, 10]. Calvanese et al. [2] 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$_g$ [13, 12]. 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. For example, Figure 1 shows a part of an EER-schema from a case study undertaken as part of the Esprit DWQ project [5, 4]. Besides cardinality restrictions and disjointness assertions (no entity is both a person and a company), we can use $\mathcal{DLR}$ axioms to explicitly axiomatise properties of entities and relations that cannot be expressed in a standard ER schema.

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 [11] have resulted in severe tractability problems when used, for example, to embed the $\mathcal{SHIF}$ description logic in $\mathcal{SHF}$ by eliminating inverse roles [14].

[1] Set semantics is assumed in this framework.

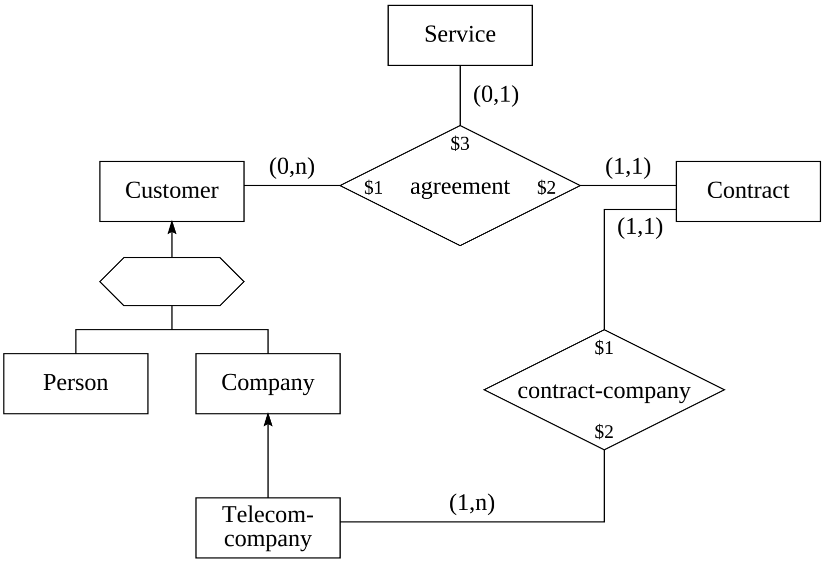

Figure 1: A fragment of the DWQ enterprise schema

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 [2], 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 [5, 4]. 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 in [2], but extends $\mathcal{DLR}$ by defining an *ABox*, i.e., a set of axioms that assert facts about named *individuals* and tuples of named individuals (see [3]). 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 [20], 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 our approach represents a practical solution. In the FaCT system [14], we already have an (optimised) implementation of the decision procedure for $\mathcal{SHIQ}$ schema satisfiability described in [17], and using FaCT we have been able to reason very efficiently with a realistic schema derived from the integration of several EER-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 [5, 4]. We have already shown (in [18]) how the $\mathcal{SHIQ}$ algorithm implemented in the FaCT system can be extended to deal with ABox axioms, and in Section 4 we use FaCT to demonstrate the empirical tractability of a simple query containment problem with respect to the integrated DWQ schema. As the number of individuals generated by the encoding of realistic query containment problems will be relatively small (i.e., equal to the number of variables in the queries), the extension to deal with arbitrary ABox axioms (and thus arbitrary query containment problems) should not compromise this empirical tractability.

Due to space limitations, most details and proofs are either omitted or given only as outlines in this paper. For full details, please refer to [16] .

2 Preliminaries

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 [6].

Definition 2.1 Given a set of atomic concepts NC and a set of atomic relations NR, every $C \in \mathsf{NC}$ is a concept and every $\mathbf{R} \in \mathsf{NR}$ is a relation, with every $\mathbf{R}$ having an associated arity. If C, D are concepts, $\mathbf{R}, \mathbf{S}$ are relations of arity n, i is an integer $1 \leqslant i \leqslant n$, and k is a non-negative integer, then

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

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

A $\mathcal{DLR}$ *schema* $\mathcal{S}$ is a set of *axioms* of the form $C \sqsubseteq D$ and $\mathbf{R} \sqsubseteq \mathbf{S}$, where C, D are $\mathcal{DLR}$ concepts and $\mathbf{R}, \mathbf{S}$ are $\mathcal{DLR}$ relations of the same arity.

Given a set of individuals NI, a $\mathcal{DLR}$ *ABox* $\mathcal{A}$ is a set of *axioms* of the form $w{:}C$ and $\vec{w}{:}\mathbf{R}$, where C is a concept, $\mathbf{R}$ is a relation of arity n, w is an individual and $\vec{w}$ is an n-tuple $\langle w_1, \ldots, w_n\rangle$ such that $w_1, \ldots, w_n$ are individuals.

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 $\Delta^{\mathcal{I}}$, every n-ary relation to a subset of $(\Delta^{\mathcal{I}})^n$, and every individual to an element in $\Delta^{\mathcal{I}}$ such that the following equations are satisfied ("$\sharp$" denotes set cardinality).

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

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 $\mathbf{R}$ with arity n is relative to $\top_n$.

An interpretation $\mathcal{I}$ *satisfies* an axiom $C \sqsubseteq D$ ($\mathbf{R} \sqsubseteq \mathbf{S}$, $w{:}C$, or $\vec{w}{:}\mathbf{R}$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($\mathbf{R}^{\mathcal{I}} \subseteq \mathbf{S}^{\mathcal{I}}$, $w^{\mathcal{I}} \in C^{\mathcal{I}}$, or $\vec{w}^{\mathcal{I}} \in \mathbf{R}^{\mathcal{I}}$). An interpretation $\mathcal{I}$ *satisfies* a schema $\mathcal{S}$ (an ABox $\mathcal{A}$) iff $\mathcal{I}$ satisfies every axiom in $\mathcal{S}$ (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 $\mathcal{I} \models X$.

Definition 2.2 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 $\langle \mathcal{S}, \mathcal{A} \rangle {\mid\!\approx} \mathcal{A}'$) iff every model $\mathcal{I}$ of $\langle \mathcal{S}, \mathcal{A} \rangle$ can be extended to $\mathcal{A}'$.

For example, translating the EER-schema from Figure 1 to $\mathcal{DLR}$ axioms yields, besides others, the following axioms (where we use $C \doteq D$ as an abbreviation for $C \sqsubseteq D$ and $D \sqsubseteq C$):

$$\begin{array}{rcl}
\text{Customer} & \doteq & \text{Company} \sqcup \text{Person} \\
\text{Company} & \doteq & \text{Customer} \sqcap \neg\text{Person} \\
\text{Telecom-company} & \sqsubseteq & \text{Company} \sqcap \exists[\$1]\text{contract-company} \\
\text{contract-company} & \doteq & \exists[\$1]\text{Contract} \sqcap \exists[\$2]\text{Telecom-company} \\
 & \vdots &
\end{array}$$

Moreover, one of the $\mathcal{DLR}$ axioms defining the relationship between the enterprise schema and the entity "Business-Customer" in a certain source schema describing business contracts is

$$\begin{array}{rl}
\text{Business-Customer} \;\sqsubseteq & (\text{Company} \sqcap \exists[\$1](\text{agreement} \sqcap \\
 & \quad (\$2/3 : (\text{Contract} \sqcap \exists[\$1](\text{contract-company} \sqcap \\
 & \quad\quad (\$2/2 : \text{Telecom-company})))))).
\end{array} \tag{A1}$$

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.

Queries: In this paper we will focus on conjunctive queries (see [1, chap. 4]). A *conjunctive query* q is an expression

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

where $\vec{x}$, $\vec{y}$, and $\vec{c}$ 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 $term_i(\vec{x},\vec{y},\vec{c})$ is called an *atom* in q and is in one of the forms $C(w)$ or $\mathbf{R}(\vec{w})$, where w (resp. $\vec{w}$) is a variable or constant (resp. tuple of variables and constants) in $\vec{x}$, $\vec{y}$ or $\vec{c}$, C is a $\mathcal{DLR}$ concept, and $\mathbf{R}$ is a $\mathcal{DLR}$ relation.[2]

Continuing the telecom example, a query designed to return those companies that have an agreement on an sms service that they provide themselves is:

$$q(x) \leftarrow \text{agreement}(x, y, \text{sms}) \wedge \text{contract-company}(y, x)$$

This query contains a distinguished variable x, one undistinguished variable y, and the constant sms.

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 $\vec{d} \in (\Delta^{\mathcal{I}})^n$ such that the first order formula $\exists\vec{y}.term_1(\vec{d},\vec{y},\vec{c}) \wedge \ldots \wedge term_n(\vec{d},\vec{y},\vec{c})$ is true in $\mathcal{I}$.

As usual, we require unique interpretation of constants, i.e., in the following we will only consider those interpretations $\mathcal{I}$ with $c^{\mathcal{I}} \neq d^{\mathcal{I}}$ for any two constants $c \neq d$. A query $q(\vec{x})$ 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 $q_1(\vec{x})$ is *contained* in a query $q_2(\vec{x})$ 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 $\mathcal{S} \models q_2 \sqsubseteq q_1$.

Again, in our running example with the schema from Figure 1 and axiom A1, it is relatively easy to see that the query

$$q_1(x) \leftarrow \text{Business-Customer}(x) \tag{Q1}$$

is contained in the query

$$\begin{aligned} q_2(x) \quad \leftarrow \quad & \text{agreement}(x, y_1, y_2) \wedge \text{Contract}(y_1) \wedge \text{Service}(y_2) \wedge \\ & \quad \text{contract-company}(y_1, y_3) \wedge \text{Telecom-company}(y_3) \end{aligned} \tag{Q2}$$

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

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.

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

Definition 2.3 Given a set of atomic concepts NC and a set of atomic roles NR with transitive role names $\mathsf{NR}_+ \subseteq \mathsf{NR}$, every $C \in \mathsf{NC}$ is a $\mathcal{SHIQ}$ 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

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

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.

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

$\mathcal{SHIQ}$ 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 [17]: if a role S occurs in a number restriction $\leqslant kS.C$, then neither S nor any of its subroles 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 $\langle \mathcal{S}, \mathcal{A}_1\rangle \models\!\!\approx \mathcal{A}_2$ (see Definition 2.2). Secondly, the ABox inclusion problem is transformed into one or more KB satisfiability problems. The last step of this reduction is given in [16], where we show how a $\mathcal{DLR}$ KB can be polynomially 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 3.1 Let q be a conjunctive query. The *canonical ABox* for q is defined by

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

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 queries q_1 and q_2, $\mathcal{S} \models q_1 \sqsubseteq q_2$ iff $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle \mathrel{|}\!\!\approx \mathcal{A}_{q_2}$.*

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, $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle \not\mathrel{|}\!\!\approx \widehat{\mathcal{A}}_{q_2}$ 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 individuals in $\mathcal{A}_{q_2}$ corresponding to non-distinguished variables without the introduction of new representative concepts.

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 $(d_1, \dots, d_n) \in (\Delta^{\mathcal{I}})^n$ such that $(d_1, \dots, d_n) \in q_1(\mathcal{I})$ and $(d_1, \dots, d_n) \notin q_2(\mathcal{I})$. $\mathcal{I}$ and the assignment of variables leading to $(d_1, \dots, d_n)$ 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 $(d_1, \dots, d_n) \in q_2(\mathcal{I})$, 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 $(d_1, \dots, d_n) \in q_1(\mathcal{I})$ 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. ■

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 $\vec{w}{:}\mathbf{R}$ 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: $\langle \mathcal{S}, \widehat{\mathcal{A}}_{q_1} \rangle \mathrel{|\!\!\approx} \mathcal{A}_{q_2}$ 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 $\vec{w}{:}\mathbf{R}$ for each tuple $\vec{w}$, 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. $\{\vec{w}{:}\mathbf{R}, \vec{w}{:}\mathbf{S}\}$) iff $\mathcal{I} \models \{w{:}(C \sqcap D)\}$ (resp. $\{\vec{w}{:}(\mathbf{R} \sqcap \mathbf{S})\}$). The following lemma shows that we can also absorb $w_i{:}C$ into $\vec{w}{:}\mathbf{R}$ when w_i is an element of $\vec{w}$.

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

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

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 $\vec{w}{:}\mathbf{R}$). Two distinct connected components do not have any individuals in common, hence we can deal independently with the inclusion problem for each connected set of axioms: $\langle \mathcal{S}, \mathcal{A} \rangle \mathrel{|\!\!\approx} \mathcal{A}'$ iff $\langle \mathcal{S}, \mathcal{A} \rangle \mathrel{|\!\!\approx} \mathcal{G}$ for every maximal connected set of axioms $\mathcal{G} \subseteq \mathcal{A}'$.

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 collapsing would be easy if we were able to refer to individuals in C (i.e., if our logic included *nominals* [21]), which is not the case. 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}. For non-distinguished individuals, things are more complicated.

- For those non-distinguished individuals z_i that are encountered only once during the traversal and hence only have to be referred to once in the collapsed axiom $w{:}C$, we will use $\top$ as their representative concept P_{z_i}.

- Those non-distinguished individuals z_i that are encountered more than once during the traversal will have to be referred to at least twice. For them we use the representative concept P_{w_j} for some individual w_j occuring in $\widehat{\mathcal{A}}_{q_1}$.

The use of $\top$ and the representative concepts P_{w_j} to refer to the non-distinguished individuals z_i in $\mathcal{G}$ can be informally justified as follows.[3] When an interpretation $\mathcal{I}$ is extended to $\mathcal{G}$, z_i can be interpreted as any element in $\Delta^{\mathcal{I}}$ $(= \top^{\mathcal{I}})$, and when z_i is referred to only once in $w{:}C$ there is no other constraint on its interpretation. When z_i is referred to more than once in $w{:}C$, we cannot use $\top$ as, although the extension of $\mathcal{I}$ can still interpret z_i as any element, we must be sure that we are referring to the same element in every case. However, due to the way we will construct $w{:}C$, z_i will only be referred to more than once if it occurs in a cycle, and $z_i^{\mathcal{I}'}$ must then be in a corresponding cycle in any model $\mathcal{I}'$ of $\mathcal{G}$. If $\langle \mathcal{S}, \mathcal{A} \rangle \models\!\!\!\approx \mathcal{G}$, then every model $\mathcal{I}$ of $\langle \mathcal{S}, \mathcal{A} \rangle$ must contain such a cycle (extending $\mathcal{I}$ to $\mathcal{G}$ cannot introduce a cycle) and, due to the properties of $\mathcal{DLR}$, this can only be guaranteed if the cycle is explicitly asserted in the axioms of $\widehat{\mathcal{A}}_{q_1}$. We can therefore assume that z_i has the same interpretation as one of the w_j in $\widehat{\mathcal{A}}_{q_1}$, and we can use P_{w_i} to refer to it. Unfortunately, at this time of the reduction, it is impossible to determine which P_{w_j} is the appropriate choice for any such z_i, and we have to rely on non-deterministic guessing.

Note that guessing a P_{w_i} for each such z_i adds non-determinism to the procedure, which has to be dealt with in a deterministic implementation. This should be manageable as the number of such z_i will typically be very small.[4] On the other hand, the apparent requirement for advanced knowledge of how many times the z_i will be referred to in the collapsed axiom $w{:}C$ is not a problem: in practice we will use "place-holder" concepts, and make the appropriate substitutions after completing the collapsing procedure.

Interestingly, also in [10], 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 [9].

[3] For full details, the reader is again referred to [16].

[4] This represents a useful refinement over the procedure described in [2], where all z_i that occur in cycles are non-deterministically replaced with one of the w_i, regardless of whether or not they are used to enforce a co-reference.

The following lemma shows how we can use the representative concepts to transform an axiom of the form $\vec{w}{:}\mathbf{R}$ 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 $\vec{w}{:}\boldsymbol{R} \in \mathcal{A}'$, then $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle \mid\!\approx \mathcal{A}'$ iff $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle \mid\!\approx (\{w_i{:}C\} \cup \mathcal{A}' \setminus \{\vec{w}{:}\boldsymbol{R}\})$, where $\vec{w} = \langle w_1, \ldots, w_n \rangle$, w_i is the ith element in $\vec{w}$, C is the concept*

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

and P_{w_j} is the appropriate concept for referring to w_j.

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 $\vec{w}{:}\mathbf{R}$, 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 $\vec{w}{:}\mathbf{R}$, 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. ■

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 $\vec{w}{:}\mathbf{R}$ into $w_i{:}C$, we can then use Lemma 1 to absorb $w_i{:}C$ into another axiom $\vec{v}{:}\mathbf{R}$, where w_i is an element of $\vec{v}$. We can do this using a recursive traversal of the graphical representation of $\mathcal{G}$ (a similar technique is used in [2] 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, $\vec{w}$, is selected, visited, and the axiom $\vec{w}{:}\mathbf{R}$ transformed into an axiom $w_i{:}C$. Finally, any axioms $w_i{:}C_1, \ldots, w_i{:}C_n$ resulting from these transformations are merged into a single axiom $w_i{:}(C_1 \sqcap \ldots \sqcap C_n)$.
- At a tuple node $\vec{w}$, the node is first marked as visited. Then, while there remains an unmarked individual node connected to $\vec{w}$, one of these, w_i, is selected, visited, and any axiom $w_i{:}C$ that results from the visit is merged into the axiom $\vec{w}{:}\mathbf{R}$ 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 $\mathcal{G}$, 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 $\langle \mathcal{S}, \widehat{\mathcal{A}} \rangle \mid\!\approx \{w{:}C\}$ 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 $\widehat{\mathcal{A}}$, $\langle\mathcal{S},\widehat{\mathcal{A}}\rangle\not\approx\{w{:}C\}$ 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 $\Delta^{\mathcal{I}} \not\subseteq (\neg C)^{\mathcal{I}}$. ■

In our running example, Query Q1 and Query Q2 are transformed into the following $\mathcal{DLR}$ ABoxes

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

In the next step, $\mathcal{A}_{q_2}$ is collapsed to $\{x{:}C'_{q_2}\}$, where

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

Now we can then 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 $\mathcal{A} = \{x{:}\text{Business-Customer}, x{:}P_x, x{:}\neg C'_{q_2}\}$. This is equivalent to testing the satisfiability of the concept Business-Customer $\sqcap P_x \sqcap \neg C'_{q_2}$ w.r.t. $\mathcal{S}$.

Summing up, in this section we have shown:

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 *all* possible mappings from the set Z of non-distinguished individuals that occur more than once in the collapsed $\mathcal{G}$ to the set W of individuals that occur in $\widehat{\mathcal{A}}_1$, of which there are $|W|^{|Z|}$ many. However, both these sets 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). Therefore, we do not believe that this additional non-determinism compromises the feasibility of our approach.

Together with the reduction of satisfiability of $\mathcal{DLR}$-ABoxes to satisfiability of $\mathcal{SHIQ}$-knowledge bases given in [16], 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 is currently being extended to deal with ABox axioms [18].

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 [2]. Satisfiability of a $\mathcal{SHIQ}$-KB can be determined in ExpTime.[5] All reduction steps can be computed in

[5] This does not follow from the algorithm presented in [18], which focuses on feasibility rather than worst-case complexity. It can be shown using a precompletion strategy similar to the one used in [22] together with the ExpTime-completeness of $\mathcal{CIQ}$ [12].

deterministic polynomial time, with the exception of the reduction used in Theorem 2, which requires the consideration of exponentially many mappings. Yet, for every fixed mapping, the reduction is polynomial, which yields that our approach decides query containment in ExpTime. A feasible algorithm to decide satisfiability of $\mathcal{SHIQ}$-KBs is given in [18].

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 the query containment problem with respect to the DWQ schema mentioned in Section 3.

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 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 [5]. A part of the *enterprise* schema which represents the global concepts and relationships that are of interest in the Data Warehouse is shown in Figure 1. 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 is given in Equation A1.

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 C is not satisfiable w.r.t. $\mathcal{S}$. Moreover, if $\mathcal{S}$ is first *classified* (i.e., the subsumption partial ordering of all named concepts in $\mathcal{S}$ 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 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. [2]. During the DWQ project, attempts were made to classify the DWQ schema using a similar embedding in the less expressive $\mathcal{SHIF}$ logic [15] 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 sketched 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 main difference is that, although each conjunctive part becomes an ABox, the object representing the whole disjunctive query is a *set* of 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 [16].

Although there is some loss of expressive power with respect to the framework presented in [2] 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.

References

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

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

[3] D. Calvanese, G. De Giacomo, and M. Lenzerini. Answering queries using views in description logics. In E. Franconi and M. Kifer, editors, *Proc. of KRDB'96.*

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

[5] 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.

[6] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Proc. of KR-98*, 1998. Morgan Kaufmann Publishers.

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

[8] Edward P. F. Chan. Containment and minimization of positive conjunctive queries in OODB's. In ACM, editor, *Proc. of PODS'92*, 1992. ACM Press.

[9] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proc. of STOC'77*, 1977. ACM Press.

[10] C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. In *Proc. of ICDT-97*, LNCS 1186. Springer-Verlag, 1997.

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

[12] G. De Giacomo and M. Lenzerini. TBox and ABox reasoning in expressive description logics. In L. C. Aiello, J. Doyle, and S. C. Shapiro, editors, *Proc. of KR'96.* Morgan Kaufmann Publishers, 1996.

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

[14] I. Horrocks. FaCT and iFaCT. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors, *Proc. of DL'99*, 1999.

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

[16] 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.

[17] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. of LPAR'99*, LNAI 1705. Springer-Verlag, 1999.

[18] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$. In *Proc. of the 17th Int. Conf. on Automated Deduction (CADE-17)*, 2000. To appear.

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

[20] A. Y. Levy and M.-C. Rousset. CARIN: A representation language combining horn rules and description logics. In W. Wahlster, editor, *Proc. of ECAI'96.* John Wiley & Sons Ltd., 1996.

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

[22] S. Tessaris and G. Gough. Abox reasoning with transitive roles and axioms. In *Proc. of DL'99.* Available from http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-22/.