Conjunctive Query Entailment for $\mathcal{SHOQ}$

Birte Glimm*, Ian Horrocks, and Ulrike Sattler

[glimm,horrocks,sattler]@cs.man.ac.uk
The University of Manchester, UK

Abstract. An important reasoning task, in addition to the standard DL reasoning services, is conjunctive query answering. In this paper, we present a decision procedure for conjunctive query entailment in the expressive Description Logic $\mathcal{SHOQ}$. In particular, we present the first query entailment decision procedure for a logic that allows for nominals. We achieve this by adapting the query rewriting and rolling-up technique that is also used in the query entailment procedure for $\mathcal{SHIQ}$.

1 Introduction

Existing Description Logic (DL) reasoners[1] provide automated reasoning support for checking concepts for satisfiability and subsumption, and also for answering queries that retrieve instances of concepts and roles. There are, however, still many open questions regarding the development of algorithms that decide conjunctive query (CQ) entailment in expressive Description Logics. A decision procedure for conjunctive query entailment in $\mathcal{SHIQ}$ is known only recently [1]. Previously proposed techniques for deciding CQ entailment in expressive DLs mostly require that all roles that occur in the query are simple, i.e., neither transitive nor have transitive subroles. Furthermore, none of the existing conjunctive query answering techniques [2–6] is able to handle nominals. In this paper, we address this issue and present a decision procedure for CQ entailment in the very expressive DL $\mathcal{SHOQ}$, i.e., we allow for both features that were problematic for previous algorithms: non-simple roles in the query and nominals.

We achieve this, by combining the ideas from the CQ entailment decision procedure for $\mathcal{SHIQ}$ [1] with the technique for deciding entailment of $\mathcal{SHOQ}$ CQs that have just one strongly connected query graph component [9]. We first rewrite a query into a set of queries that have a kind of forest shape. We then use the rolling-up or tuple-graph technique [4, 2] in order to build concepts that capture the rewritten queries. We can then use the obtained concepts for reducing the task of deciding query entailment to the task of testing the consistency of extendeded knowledge bases.

* This work was supported by an EPSRC studentship.

[1] For example, FaCT++ http://owl.man.ac.uk/factplusplus, KAON2 http://kaon2.semanticweb.org, Pellet http://pellet.owldl.com, or Racer Pro http://www.racer-systems.com

2 Preliminaries

We assume readers to be familiar with the syntax and semantics of the DL $\mathcal{SHOQ}$ (for details see [7]). Since in the presence of nominals the ABox can be internalised, we assume that a $\mathcal{SHOQ}$ knowledge base $\mathcal{K}$ is a tuple $(\mathcal{T}, \mathcal{R})$ over a signature $\mathcal{S} = (N_C, N_R)$, where $\mathcal{T}$ is a TBox, $\mathcal{R}$ is a role hierarchy, and N_C and N_R are countable, infinite, and pairwise disjoint sets of *concept names* and *role names* respectively. We assume that the set of concept names contains a subset N_I of nominal names and the set N_R contains a subset $\mathsf{N_{tR}}$ of *transitive role names*. We use $\mathsf{nom}(\mathcal{K})$ for the set of nominals that occur in $\mathcal{K}$ and we say that a role r is simple if there is no $s \in \mathsf{N_{tR}}$ such that $s \sqsubseteq^*_{\mathcal{R}} r$, where $\sqsubseteq^*_{\mathcal{R}}$ is the reflexive transitive closure of $\sqsubseteq$ over $\mathcal{R}$.

Definition 1. *Let $\mathcal{S}$ be a signature and N_V a countably infinite set of variable names disjoint from N_C and N_R. Let C be a $\mathcal{SHOQ}$-concept over $\mathcal{S}$, $r \in N_R$ a role name, and $x, y \in N_V$. An atom is an expression $C(x)$ or $r(x, y)$ and we refer to these types of atoms as concept atoms and role atoms respectively. A* Boolean conjunctive query *q is a non-empty set of atoms. We use $\mathsf{Vars}(q)$ to denote the set of variables occurring in q and $\sharp(q)$ for the cardinality of q.*

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation. For a total function $\pi \colon \mathsf{Vars}(q) \to \Delta^{\mathcal{I}}$, we write

- $\mathcal{I} \models^{\pi} C(x)$ *if* $\pi(x) \in C^{\mathcal{I}}$;
- $\mathcal{I} \models^{\pi} r(x, y)$ *if* $(\pi(x), \pi(y)) \in r^{\mathcal{I}}$.

If $\mathcal{I} \models^{\pi} at$ for all atoms $at \in q$, we write $\mathcal{I} \models^{\pi} q$. We say that $\mathcal{I}$ satisfies *q and write $\mathcal{I} \models q$ if there exists a mapping π such that $\mathcal{I} \models^{\pi} q$. We call such a π a* match *for q in $\mathcal{I}$. For a $\mathcal{SHOQ}$ knowledge base $\mathcal{K}$, we say that $\mathcal{K}$* entails *q and write $\mathcal{K} \models q$ if $\mathcal{I} \models \mathcal{K}$ implies $\mathcal{I} \models q$.*

Please note that we do not allow for constants (individual names) in the query. In the presence of nominals this is clearly w.l.o.g. Since answering non-Boolean conjunctive queries can be reduced to answering (possibly several) Boolean queries, we consider only Boolean queries here.

In the following, we use $\mathcal{K}$ for a $\mathcal{SHOQ}$ knowledge base and q for a Boolean conjunctive query over a common signature $\mathcal{S}$.

As for the CQ entailment algorithm for $\mathcal{SHIQ}$, we first show that we can restrict our attention to the canonical models of $\mathcal{K}$, which are models that have a kind of forest shape.

Definition 2. *A* tree *T is a prefix-closed subset of $\mathbb{N}^*$. For $w, w' \in T$, we call w' a* successor *of w if $w' = w \cdot c$ for some $c \in \mathbb{N}$, where "$\cdot$" denotes concatenation. The empty word ε is the root.*

A forest base for *$\mathcal{K}$ is an interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ that interprets transitive roles in an unrestricted (i.e., not necessarily transitive) way and, additionally, satisfies the following conditions:*

T1 *$\Delta^{\mathcal{J}} \subseteq \mathsf{nom}(\mathcal{K}) \times \mathbb{N}^*$ such that, for all $o \in \mathsf{nom}(\mathcal{K})$, the set $\{w \mid (o, w) \in \Delta^{\mathcal{J}}\}$ is a tree;*

T2 *if* $((o,w),(o',w')) \in r^{\mathcal{J}}$*, then either* $w' = \varepsilon$ *or* $o = o'$ *and* w' *is a successor of* w*;*

T3 *for each* $o \in \mathsf{nom}(\mathcal{K})$*,* $o^{\mathcal{J}} = (o, \varepsilon)$*;*

An interpretation $\mathcal{I}$ *is* canonical for $\mathcal{K}$ *if there exists a forest base* $\mathcal{J}$ *for* $\mathcal{K}$ *such that* $\mathcal{I}$ *is identical to* $\mathcal{J}$ *except that, for all non-simple roles* r*, we have*

$$r^{\mathcal{I}} = r^{\mathcal{J}} \cup \bigcup_{s \sqsubseteq_{\mathcal{R}} r,\ s \in \mathsf{N_{tR}}} (s^{\mathcal{J}})^{+}$$

In this case, we say that $\mathcal{J}$ *is a forest base* for $\mathcal{I}$ *and that* $\mathcal{I}$ *is a* canonical model for $\mathcal{K}$*.*

The following lemma motivates our focus on canonical models.

Lemma 1. *$\mathcal{K} \not\models q$ iff there is some canonical model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \not\models q$.*

Proof Sketch: The if direction is trivial. For the only if direction, the proof is similar to the one for $\mathcal{SHIQ}$. Let $\mathcal{I}$ be such that $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \not\models q$. Intuitively, we first unravel $\mathcal{I}$ into a model $\mathcal{I}'$ of $\mathcal{K}$ and then construct a forest base from the unravelled model. Finally, we obtain a canonical model from the forest base by transitively closing all roles $r \in \mathsf{N_{tR}}$. Since in the unravelling process we only "break" cycles, the query is still not satisfied in the constructed canonical model.

3 Reducing Query Answering to Concept Unsatisfiability

In this section, we introduce more intuitively than precisely the basic concepts that have been used in the development of algorithms for CQ entailment. In the following section, we show more formally how the ideas and techniques presented here can be combined in order to obtain a decision procedure for $\mathcal{SHOQ}$.

The initial ideas used in this paper were first introduced by Calvanese et al. [4] for deciding conjunctive query containment and hence CQ entailment for $\mathcal{DLR}_{reg}$. The authors show how a query q can be expressed as a concept C_q, such that q is true w.r.t. a given knowledge base if adding $\top \sqsubseteq \neg C_q$ makes the KB inconsistent. In order to obtain the concept C_q, the query q is represented as a directed, labelled graph. This graph, called a tuple graph or a query graph, is traversed in a depth-first manner and, during the traversal, nodes and edges are replaced with appropriate concept expressions, leading to the concept C_q after completing the traversal.

The nodes in a query graph correspond to the terms in the query and are labelled with the concepts that occur in the corresponding concept atoms. The edges correspond to the role atoms in q and are labelled accordingly. For example, let $q_1 = \{C(x), s(x,y), D(y)\}$ and $q_2 = \{C(x), r(x,y), r(x,y'), s(y,z), s(y',z), D(z)\}$. The query graphs for q_1 and q_2 are depicted in Fig. 1 and Fig. 2 respectively. We call q_2 a cyclic query since its underlying undirected query graph is cyclic. Since q_1 is acyclic, we can build the concept that represents q_1 as follows: start at x

and traverse the graph to y. Since y is a leaf node, remove y and its incoming edge and conjoin $\exists s.D$ to the label C of x, resulting in $C \sqcap \exists s.D$ for C_{q_1}. A given KB $\mathcal{K}$ entails q_1 iff $\mathcal{K} \cup \{\top \sqsubseteq \neg C_{q_1}\}$ is inconsistent.

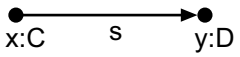

Fig. 1: The (acyclic) query graph for q_1.

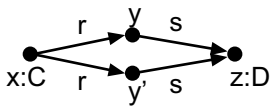

Fig. 2: A query graph for the cyclic query $q?2$.

This reduction is not directly extendable to cyclic queries since, due to the tree model property of most DLs, a concept cannot capture cyclic relationships. We can, however, also not simply replace variables in a cycle with individual names from the ABox, where arbitrary cyclic structures can be expressed, since by identifying variables with each other some cyclic queries become acyclic. For example, identifying y and y' in q_2 leads to an acyclic query.

Last year, we presented an algorithm for conjunctive query entailment in $\mathcal{SHOQ}$ that is a decision procedure for queries for which the corresponding query graph consists of one strongly connected component (i.e., we can reach each node from each other node in the directed query graph) [9]. In the presence of nominals, a simple non-deterministic assignment of nominals to variables that occur in a cycle is not sufficient even after identifying variables with each other. For example, Fig. 3 represents a model for the KB containing the axioms $\{a\} \sqsubseteq \neg C \sqcap \neg D \sqcap \exists s.(C \sqcap \exists r.(D \sqcap \exists s.\{a\}))$ with $s \in \mathsf{N_{tR}}$. The query $\{C(x), D(y), r(x,y), s(y,x)\}$ would clearly be satisfied in each model of $\mathcal{K}$, although in the relevant matches neither x nor y can be mapped to the nominal $a^{\mathcal{I}}$.

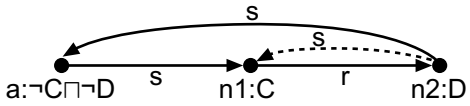

Fig. 3: The dashed line indicates the relationship added due to s being transitive. Therefore, there is a cycle not directly containing the nominal a.

In canonical models for a $\mathcal{SHOQ}$ KB, a directed cycle among non-nominals can only occur due to a transitive role that provides a shortcut for "skipping" the nominal. Hence, a nominal is always at least indirectly involved, e.g., a in the current example. The proposed algorithm allows, therefore, the replacement of the role atom $s(y,x)$ with two role atoms $s(y,v), s(v,x)$ for a new variable v. We can then guess that v corresponds to the nominal a and express the

query as a concept in which we use the nominal a to close the cycle. In order to decide CQ entailment, we build a set of queries by identifying variables with each other and by replacing role atoms with non-simple roles as described above. We then express all queries as concepts by identyfying variables in the cycles with nominals in all possible ways where necessary to close the cycle when rolling-up the query into a concept. We then check whether $\mathcal{K}$ entails the disjunction of the obtained concepts.

In the decision procedure for CQ entailment in $\mathcal{SHIQ}$ [1], a given query is also rewritten into a set of ground or tree-shaped CQs such that the disjunction of these rewritten queries can be used to decide CQ entailment. The rewriting steps for $\mathcal{SHIQ}$ also allow for eliminating shortcuts induced by transitive roles that do not involve nominals (or ABox individual in the case of $\mathcal{SHIQ}$). For example, the query $\{t(x,y), t(y,z), t(x,z)\}$ for $t \in \mathsf{N}_{\mathsf{tR}}$ is true iff the query $\{t(x,y), t(y,z)\}$ is true. In the rewriting process we do not directly delete the "shortcut" $t(x,z)$, but explicate it by replacing $t(x,z)$ with two role atoms $t(x,y), t(y,z)$, i.e., this time we reuse the variable y instead of introducing a new variable.

We now show how we can combine these techniques in order to obtain a decision procedure for general CQs in $\mathcal{SHOQ}$.

4 Conjunctive Query Entailment for $\mathcal{SHOQ}$

For deciding whether a given Boolean CQ is entailed by a $\mathcal{SHOQ}$ KB, we transform the query in a four stage process into a set of $\mathcal{SHOQR}$ concepts, i.e., $\mathcal{SHOQ}$ with role conjunctions. We can then reduce the task of deciding CQ entailment to the task of deciding KB consistency.

In the first step, called collapsing, we can identify variables with one another. In the second step, we can replace role atoms of the form $r(x,x')$ for which r is non-simple with up to $\sharp(q)$ role atoms. This allows for explication all shortcuts in the query. In the third step, we decide which variables correspond to nominals and filter out those queries that can still not be expressed as a $\mathcal{SHOQR}$ concept. Those queries are trivially false since the structure specified by the query cannot be enforced by a $\mathcal{SHOQ}$ concept and cannot be mapped to the canonical models of the KB. Finally, we express the resulting queries as concepts and show how we can use these concepts for deciding query entailment.

Definition 3. *A* collapsing *of q is obtained by identifying variables in q. We use $\mathsf{co}(q)$ to denote the set of all queries that are a collapsing of q plus q itself. A* transitivity rewriting *of q is obtained by fixing a set $V \subseteq N_V$ of variables not occurring in q such that $\sharp(V) \leqslant \sharp(q)$ and by choosing, for each role atom $r(x,x') \in q$ such that there is a role $s \in \mathsf{N}_{\mathsf{tR}}$ and $s \sqsubseteq^*_{\mathcal{R}} r$ to either*

1. *do nothing, or*
2. *replace $r(x,x')$ with $\ell \leqslant \sharp(q)$ role atoms $s(x_1,x_2), \ldots, s(x_{\ell-1},x_\ell)$, where $x_1 = x$, $x_\ell = x'$, and $x_2, \ldots, x_\ell \in \mathsf{Vars}(q) \cup V$.*

We use $\mathsf{tr}_{\mathcal{K}}(q)$ to denote the set of all queries that are a transitivity rewriting of a query $q_{co} \in \mathsf{co}(q)$.

We assume that $\mathsf{tr}_{\mathcal{K}}(q)$ contains no isomorphic queries, i.e., differences in (newly introduced) variable names only are neglected.

We now show how we can filter out those queries that are trivially false since they have a structure that cannot occur in canonical models. For this, we use forest structures that are similar to canonical models. We first decide which variables of the query correspond to nominals. Between those variables, the role atoms of the query can induce arbitrary relational structures. All other variables are mapped to trees such that for a role atom $r(x, y)$ either the image of y is a successors of the image of x in the tree or y corresponds to a nominal and $r(x, y)$ corresponds to a link to some nominal.

Definition 4. *A query q is* tree-shaped *if there exists a total and bijective function f from $\mathsf{Vars}(q)$ to a tree T such that $r(x, x') \in q$ implies that $f(x')$ is a successor of $f(x)$.*

Let $V_o \subseteq \mathsf{Vars}(q)$ be a subset of variables from q. A query forest for q w.r.t. *V_o is a set $F \subseteq V_o \times \mathbb{N}^*$ such that, for all $v_o \in V_o$, the set $\{w \mid (v_o, w) \in F\}$ is a tree.*

A query q is forest-shaped w.r.t. *V_o if either $V_o = \emptyset$ and q is tree-shaped or there is a query forest F w.r.t. V_o and a total function $f \colon \mathsf{Vars}(q) \to F$ such that*

- *if $v_o \in V_o$, then $f(v_o) = (v_o, \varepsilon)$,*
- *if $r(v, v') \in q$ and $v, v' \notin V_o$, then there is some $v_o \in V_o$ such that $f(v) = (v_o, w), f(v') = (v_o, w')$ and w' is a successor of w,*
- *if $r(v_o, v) \in q, v_o \in V_o$ and $v \notin V_o$, then $f(v) = (v_o, c)$ for $c \in \mathbb{N}$.*

We use $\mathsf{fr}_{\mathcal{K}}(q)$ to denote the set of all tuples (q_{tr}, V_o) such that $q_{tr} \in \mathsf{tr}_{\mathcal{K}}(q)$ and q_{tr} is forest-shaped w.r.t. V_o.

Similarly to forest-shaped queries, we define forest-shaped matches on canonical models.

Definition 5. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a canonical model for $\mathcal{K}$ such that $\mathcal{I} \models^{\pi} q$ for a match π. We call π a* forest match *if, for all $r(v, v') \in q$, one of the following holds:*

1. *$\pi(v') = (a, \varepsilon)$ for some $a \in \mathsf{nom}(\mathcal{K})$ or*
2. *if $\pi(v') = (a, w')$ for $w' \neq \varepsilon$, then $\pi(v) = (a, w), w'$ is a proper prefix of w, and there is no $v'' \in \mathsf{Vars}(q)$ such that $\pi(v'') = (a, w'')$ and w is a proper prefix of w'' and w'' is a proper prefix of w'.*

The following lemma shows that we can indeed omit queries that are not forest-shaped w.r.t. some subset of variables V_o.

Lemma 2. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a model for $\mathcal{K}$.*

1. *If $\mathcal{I}$ is canonical and $\mathcal{I} \models q$, then there is a tuple $(q_{tr}, V_o) \in \mathsf{fr}_{\mathcal{K}}(q)$ and a forest match π such that $\mathcal{I} \models^{\pi} q_{tr}$ and, for each $v_o \in V_o$, there is some $(a, \varepsilon) \in \Delta^{\mathcal{I}}$ such that $\pi(v_o) = (a, \varepsilon)$.*
2. *If $(q_{tr}, V_o) \in \mathsf{fr}_{\mathcal{K}}(q)$ and $\mathcal{I} \models q_{tr}$, then $\mathcal{I} \models q$.*

The proof is very similar to the proofs for $\mathcal{SHIQ}$ [1]. Intuitively, we use the canonical model $\mathcal{I}$ to guide the rewriting process in the proof of Claim 1 and Claim 2 follows from the fact that we only use non-simple roles in the transitivity rewritings.

We now build a query that consists of only concept atoms for each $(q_{tr}, V_o) \in \mathsf{fr}_{\mathcal{K}}(q)$ by replacing the variables from V_o with nominals from $\mathsf{nom}(\mathcal{K})$ and applying the rolling-up technique.

Definition 6. *Let $(q_{tr}, V_o) \in \mathsf{fr}_{\mathcal{K}}(q)$. A* grounding for q_{tr} w.r.t. V_o *is a total function $\tau\colon V_o \to \mathsf{nom}(\mathcal{K})$. For a mapping $f\colon \mathsf{Vars}(q) \to F$ to a query forest F for q_{tr} w.r.t. V_o, we build $\mathsf{con}((q_{tr}, V_o, \tau))$ as follows:*

1. *For each $r(v, v_o) \in q_{tr}$ with $v_o \in V_o$, replace $r(v, v_o)$ with $(\exists r.\{\tau(v_o)\})(v)$.*
2. *For each $v_o \in V_o$ add a concept atom $(\{\tau(v_o)\})(v_o)$ to q_{tr}.*
3. *We now inductively assign to each $v \in \mathsf{Vars}(q_{tr})$ a concept $\mathsf{con}(v)$ as follows:*
   - *if there is no role atom $r(v, v') \in q_{tr}$, then $\mathsf{con}(v) := \bigsqcap_{C(v) \in q_{tr}} C$,*
   - *if there are role atoms $r(v, v_1), \ldots, r(v, v_k) \in q_{tr}$, then*

$$\mathsf{con}(v) := \textstyle\bigsqcap_{C(v) \in q_{tr}} C \sqcap \bigsqcap_{1 \leqslant i \leqslant k} \exists \big(\bigsqcap_{r(v, v_i) \in q_{tr}} r \big).\mathsf{con}(v_i).$$

4. *Finally, $\mathsf{con}((q_{tr}, V_o, \tau)) = \{(\mathsf{con}(v))(v) \mid v \in \mathsf{Vars}(q_{tr})$ and there is no role atom $r(v', v) \in q_{tr}\}$.*

We use $\mathsf{con}_{\mathcal{K}}(q)$ for the set $\{\mathsf{con}((q_{tr}, V_o, \tau)) \mid (q_{tr}, V_o) \in \mathsf{fr}_{\mathcal{K}}(q)$ and τ a grounding w.r.t. $V_o\}$.

Please note that $\mathsf{con}((q_{tr}, V_o, \tau))$ has the form $\{C_1(x_1), \ldots, C_n(x_n)\}$ with $x_i \neq x_j$ for $1 \leq i < j \leq n$ and C_i $\mathcal{SHOQR}$-concepts.

Lemma 3. *Let $\mathcal{I}$ be a model of $\mathcal{K}$.*

1. *If $\mathcal{I}$ is canonical and $\mathcal{I} \models q$, then there is some $\mathsf{con}((q_{tr}, V_o, \tau)) \in \mathsf{con}_{\mathcal{K}}(q)$ such that $\mathcal{I} \models \mathsf{con}((q_{tr}, V_o, \tau))$.*
2. *If $\mathcal{I} \models \mathsf{con}((q_{tr}, V_o, \tau))$ for some $\mathsf{con}((q_{tr}, V_o, \tau)) \in \mathsf{con}_{\mathcal{K}}(q)$, then $\mathcal{I} \models q$.*

Intuitively, the use of nominals in the constructed concepts still enforces the same structures that are required by the query.

Putting everything together, we get the following theorem, which shows that the queries in $\mathsf{con}_{\mathcal{K}}(q)$ are indeed enough to decide whether $\mathcal{K} \models q$.

Theorem 1. *Let $\{q_1, \ldots, q_\ell\} = \mathsf{con}_{\mathcal{K}}(q)$, then $\mathcal{K} \models q$ iff $\mathcal{K} \models q_1 \vee \ldots \vee q_\ell$.*

Please note that a disjunct q_i contains possibly several concept atoms of the form $C_1^i(x_1), \ldots, C_n^i(x_n)$, i.e., n unconnected components. By transforming the disjunction into conjunctive normal form (cf. [2, 7.3.2]), we can reduce the problem of deciding whether $\mathcal{K} \models q_1 \vee \ldots \vee q_\ell$ to deciding the problem whether $\mathcal{K} \models at_1 \vee \ldots \vee at_\ell$, where each at_i is a concept atom from q_i, for at most exponentially many times in the size of the longest q_i. We now show how we can decide entailment of unions of conjunctive queries that consist of one concept atom only, which is enough to decide CQ entailment for $\mathcal{SHOQ}$.

Definition 7. *An* extended TBox w.r.t. $\mathcal{K}$ and q *is a TBox that contains, for each* $q_i \in \mathsf{con}_{\mathcal{K}}(q)$*, an axiom* $\top \sqsubseteq \neg C$ *for some atom* $C(x) \in q_i$*. An* extended knowledge base w.r.t. $\mathcal{K}$ and q *is a knowledge base* $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{R})$ *such that* $\mathcal{T}_q$ *is an extended TBox w.r.t.* $\mathcal{K}$ *and* q*.*

Theorem 2. $\mathcal{K} \models q$ *iff each extended knowledge* $\mathcal{K}'$ *w.r.t.* $\mathcal{K}$ *and* q *is inconsistent.*

Please note that the extended knowledge base is in $\mathcal{SHOQ}$ with role conjunctions (under universal quantifiers). It is, however, not hard to see how the Tableaux algorithm for $\mathcal{SHOQ}$ [10] can be extended to handle this.

5 Conclusions

In the previous section, we have presented a decision procedure for CQ entailment in $\mathcal{SHOQ}$. This is, to the best of our knowledge, the first CQ entailment decision procedure that can handle nominals. In addition, we allow for non-simple roles in the query as well, which is a feature that is known to be tricky. Since the set of rewritten queries can potentially be large, the algorithm is more suitable for showing decidability of the problem rather than building the foundation of implementable algorithms. Our future work will include efforts to show that answering arbitrary conjunctive queries for $\mathcal{SHOIQ}$ is decidable.

References

1. Glimm, B., Horrock, I., Lutz, C., Sattler, U.: Conjunctive query answering in the description logic $\mathcal{SHIQ}$. In: Proc. of IJCAI 2007. (2007)
2. Tessaris, S.: Questions and answers: reasoning and querying in Description Logic. PhD thesis, University of Manchester (2001)
3. Levy, A.Y., Rousset, M.C.: CARIN: A representation language combining horn rules and description logics. In: Proc. of ECAI 1996. (1996)
4. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: Proc. of PODS 1998. (1998)
5. Hustadt, U., Motik, B., Sattler, U.: A decomposition rule for decision procedures by resolution-based calculi. In: Proc. of LPAR 2004. (2004)
6. Ortiz, M.M., Calvanese, D., Eiter, T.: Characterizing data complexity for conjunctive query answering in expressive description logics. In: Proc. of AAAI 2006. (2006)
7. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook. (2003)
8. Horrocks, I., Sattler, U., Tessaris, S., Tobies, S.: How to decide query containment under constraints using a description logic. In: Proc. of LPAR 2000. (2000)
9. Glimm, B., Horrocks, I., Sattler, U.: Conjunctive query answering for description logics with transitive roles. In: Proc. DL 2006. (2006)
10. Horrocks, I., Sattler, U.: Ontology reasoning in the $\mathcal{SHOQ}$ description logic. In: Proc. of IJCAI 2001. (2001)