A Tractable Approach to ABox Abduction over Description Logic Ontologies

Jianfeng Du
Guangdong University of Foreign Studies,
Guangzhou 510006, China
jfdu@gdufs.edu.cn

Kewen Wang
Griffith University,
Brisbane, QLD 4111, Australia
k.wang@griffith.edu.au

Yi-Dong Shen
State Key Laboratory of Computer Science,
Institute of Software,
Chinese Academy of Sciences,
Beijing 100190, China
ydshen@ios.ac.cn

Abstract

ABox abduction is an important reasoning mechanism for description logic ontologies. It computes all minimal *explanations* (sets of ABox assertions) whose appending to a consistent ontology enforces the entailment of an observation while keeps the ontology consistent. We focus on practical computation for a general problem of ABox abduction, called the *query abduction problem*, where an observation is a Boolean conjunctive query and the explanations may contain fresh individuals neither in the ontology nor in the observation. However, in this problem there can be infinitely many minimal explanations. Hence we first identify a class of TBoxes called *first-order rewritable TBoxes*. It guarantees the existence of finitely many minimal explanations and is sufficient for many ontology applications. To reduce the number of explanations that need to be computed, we introduce a special kind of minimal explanations called *representative explanations* from which all minimal explanations can be retrieved. We develop a tractable method (in data complexity) for computing all representative explanations in a consistent ontology. Experimental results demonstrate that the method is efficient and scalable for ontologies with large ABoxes.

Introduction

In artificial intelligence, abductive reasoning (Eiter and Gottlob 1995) is an important logic-based mechanism for computing explanations for an observation that is not entailed by a background theory, where an explanation is a set of facts that should be added to the theory to enforce the entailment. Many ontology applications require to explain why an observation cannot be entailed by an ontology. As a result, abductive reasoning has regained much attention in description logics (DLs), which underpin the standard Web Ontology Language (OWL). A DL ontology consists of a TBox storing intensional information and an ABox storing extensional information. ABox abduction (Klarman, Endriss, and Schlobach 2011; Du et al. 2011a) is an adaptation of abductive reasoning to DLs. It computes all minimal *explanations* (sets of ABox assertions) whose appending to a given consistent DL ontology enforces the entailment of an observation while keeps the ontology consistent.

ABox abduction was advocated in (Elsenbroich, Kutz, and Sattler 2006) where some use scenarios such as medical diagnosis were presented. It was emphasized in (Bada, Mungall, and Hunter 2008) as a feature of support tools for ontology quality control. It was also adapted to semantic matchmaking by treating requests as observations and offers as an ABox (Du et al. 2011b). Recently, ABox abduction was applied to explain why a tuple is not an answer to a conjunctive query (Calvanese et al. 2013). As was pointed out in (Borgida, Calvanese, and Rodriguez-Muro 2008), this facility is as important as explaining why a tuple is a query answer in ontology-based data access (OBDA) systems.

Although ABox abduction is important and useful, the problem of computing minimal explanations for ABox abduction is rarely investigated. An initial attempt to compute all minimal explanations was proposed in (Klarman, Endriss, and Schlobach 2011), where a given ontology is expressed in the DL $\mathcal{ALC}$ and a given observation is a set of ABox assertions. Since the explanations can be over arbitrary $\mathcal{ALE}$ concepts, the number of minimal explanations can be infinite and the computational method proposed in (Klarman, Endriss, and Schlobach 2011) cannot guarantee termination. For example, consider an ontology consisting of a single axiom $\exists \mathsf{hasParent}.\mathsf{Person} \sqsubseteq \mathsf{Person}$, which says that something having a person as its parent is a person (Du et al. 2011a). Then the observation that Tom is a person (written $\{\mathsf{Person}(\mathsf{Tom})\}$) can have infinitely many minimal explanations of the form $\{\exists \mathsf{hasParent}. \cdots . \exists \mathsf{hasParent}.\mathsf{Person}(\mathsf{Tom})\}$. To address the termination problem, Du et al. (2011a) considered ABox abduction from a more practical perspective. They require the explanations to be over a finite set of concept names and role names (called *abducible predicates*) and to contain individuals in the given ontology only so that all minimal explanations can be computed in finite time. They also propose a method for computing all such minimal explanations in DLs that can be more expressive than $\mathcal{ALC}$.

Recently, Calvanese et al. (2013) considered the complexity aspect of a general problem of ABox abduction, called the *query abduction problem*. This problem takes Boolean conjunctive queries (BCQs) as observations and allows fresh individuals neither in the ontology nor in the observation to appear in explanations. For expressing observations, BCQs are more general than sets of ABox assertions since a set of ABox assertions can be treated as a BCQ. Allowing fresh individuals can bring more intuitive explanations. Take the aforementioned ontology for example. Suppose both $\mathsf{hasParent}$ and Tom are abducible predicates. According to the definition of ABox abduction proposed in (Du et al. 2011a), the observation $\{\mathsf{Person}(\mathsf{Tom})\}$ has only one minimal explanation $\mathcal{E} = \{\mathsf{hasParent}(\mathsf{Tom}, \mathsf{Tom}), \mathsf{Person}(\mathsf{Tom})\}$. By the definition of the query abduction problem, the same observation has some other minimal explanations that are more intuitive, such as $\mathcal{E}' = \{\mathsf{hasParent}(\mathsf{Tom}, u), \mathsf{Person}(u)\}$ where u denotes a fresh individual. $\mathcal{E}'$ is more intuitive than $\mathcal{E}$ since Tom does not necessarily have a parent who is Tom itself. On the other side, allowing fresh individuals may lead to infinitely many minimal explanations. For the above example, the observation $\mathsf{Person}(\mathsf{Tom})$ actually has infinitely many minimal explanations of the form $\{\mathsf{hasParent}(\mathsf{Tom}, u_1), \mathsf{hasParent}(u_1, u_2), \ldots, \mathsf{hasParent}(u_{n-1}, u_n), \mathsf{Person}(u_n)\}$, where $u_1, \ldots, u_n$ are fresh individuals. This makes the issue of computing all minimal explanations for the query abduction problem much tougher than that for the problem of ABox abduction addressed in (Du et al. 2011a).

Towards practical computation for the query abduction problem, we first identify a class of TBoxes called *first-order rewritable TBoxes*. We show that it guarantees the existence of finitely many minimal explanations and is sufficient for many ontology applications, such as the OBDA systems. Moreover, when the TBox is first-order rewritable, the set of minimal explanations for a BCQ can be computed in polynomial time in terms of *data complexity*, i.e. the complexity measured in the size of the ABox only.

However, the set of minimal explanations for the query abduction problem with a first-order rewritable TBox can still be too large to be computed. Consider an ontology having n axioms $\exists r_1.\top \sqsubseteq A_1, \ldots, \exists r_n.\top \sqsubseteq A_n$ in the TBox and m assertions $A_1(a_1), \ldots, A_1(a_m)$ in the ABox. Suppose all role names in the ontology are abducible predicates, then the BCQ $\{A_1(a), \ldots, A_n(a)\}$ will have $(m+1)^n$ minimal explanations of the form $\{r_1(a, u_1), \ldots, r_n(a, u_n)\}$, where u_i is an individual in $\{a_1, \ldots, a_m\}$ or a fresh individual. To reduce the number of explanations that need to be computed, we propose *representative explanations* that are minimal explanations not *strictly subsumed by* other minimal explanations. A minimal explanation $\mathcal{E}$ is said to be strictly subsumed by another one $\mathcal{E}'$ if $\mathcal{E}'$ can become a subset of $\mathcal{E}$ by replacing fresh individuals with existing or fresh individuals, but cannot vise versa. The number of representative explanations can be much smaller than that of minimal explanations. For the above example, the BCQ $\{A_1(a), \ldots, A_n(a)\}$ has only one representative explanation $\{r_1(a, u_1), \ldots, r_n(a, u_n)\}$, where $u_1, \ldots, u_n$ are different fresh individuals. Moreover, we show that the set of minimal explanations can be retrieved from the set of representative explanations.

We propose a tractable method (in data complexity) for computing all representative explanations in a consistent ontology whose TBox is first-order rewritable. It does not need to compute all minimal explanations beforehand. We compare this method with the state-of-the-art method for computing all minimal explanations (Du et al. 2011a). Experimental results show that, when both methods compute the same set of explanations, the proposed method is much more efficient and more scalable; moreover, when role names are used as abducible predicates, the computation of all minimal explanations is often impractical, but the set of representative explanations can still be efficiently computed.

Due to the space limitation, proofs in this paper are only provided in our technical report (Du, Wang, and Shen 2014).

Preliminaries

We assume that the reader is familiar with DLs (Baader et al. 2003). We only recall that a DL ontology consists of a TBox and an ABox, where the TBox contains axioms declaring the relations between concepts and roles, such as concept inclusion axioms $C \sqsubseteq D$ and role inclusion axioms $r \sqsubseteq s$, and the ABox contains assertions declaring the membership relations between individuals and concepts or roles as well as (in)equivalence relations among individuals.

We assume that the *Unique Name Assumption* (Baader et al. 2003) is adopted and only consider ABoxes consisting of *basic assertions*, namely concept assertions of the form $A(a)$ and role assertions of the form $r(a, b)$, where A is a concept name, r is a role name, and a and b are individuals. Other concept assertions and role assertions can be normalized to basic ones in a standard way. Let Σ be a set of concept names and role names. An ABox that contains only concept names and role names from Σ is called a Σ-ABox.

We use the traditional semantics for DLs given e.g. in (Baader et al. 2003). A DL ontology $\mathcal{O}$ is said to be *consistent*, denoted by $\mathcal{O} \not\models \bot$, if it has at least one model; otherwise, it is *inconsistent*, denoted by $\mathcal{O} \models \bot$.

A Boolean conjunctive query (BCQ) is of the form $\exists \vec{x}\, \Phi(\vec{x}, \vec{c})$, where $\Phi(\vec{x}, \vec{c})$ is a conjunction of atoms over concept names and role names, where $\vec{x}$ are variables and $\vec{c}$ are individuals. A BCQ is written as and can also be treated as a set of atoms. For example, the BCQ $\exists x\, A(a) \wedge r(a, x)$ is written as $\{A(a), r(a, x)\}$. A *substitution* for a BCQ Q is a mapping from variables in Q to individuals or variables; it is called *ground* if it maps variables in Q to individuals only. A BCQ Q is said to be *entailed by* a DL ontology $\mathcal{O}$ if Q is satisfied by all models of $\mathcal{O}$, written $\mathcal{O} \models Q$.

Some DLs can be translated to Datalog$^{\pm}$ (Calì, Gottlob, and Lukasiewicz 2012). A Datalog$^{\pm}$ ontology consists of finitely many *tuple generating dependencies* (TGDs) $\forall \vec{x} \forall \vec{y}\, \Phi(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}\, \varphi(\vec{x}, \vec{z})$, *negative constraints* (simply *constraints*) $\forall \vec{x}\, \Phi'(\vec{x}) \rightarrow \bot$, as well as *equality generating dependencies* (EGDs) $\forall \vec{x}\, \Phi'(\vec{x}) \rightarrow x_1 = x_2$, where $\Phi(\vec{x}, \vec{y})$, $\varphi(\vec{x}, \vec{z})$ and $\Phi'(\vec{x})$ are conjunctions of atoms, x_1 and x_2 occur in $\vec{x}$, and $\bot$ denotes the truth constant false. The portions of a TBox $\mathcal{T}$ that are translated to TGDs, constraints and EGDs are denoted by $\mathcal{T}_D$, $\mathcal{T}_C$ and $\mathcal{T}_E$, respectively. We introduce the notion of *first-order rewritable TBox* below.

Definition 1. A TBox $\mathcal{T}$ is said to be *first-order rewritable* if it can be translated to a Datalog$^{\pm}$ ontology and satisfies the following three conditions for an arbitrary BCQ Q and an arbitrary Σ-ABox $\mathcal{A}$, where Σ is the set of concept names and role names in $\mathcal{T}$:

(1) $\mathcal{T} \cup \mathcal{A} \models Q$ if and only if $\mathcal{T}_D \cup \mathcal{A} \models Q$ or $\mathcal{T} \cup \mathcal{A} \models \bot$;

(2) $\mathcal{T}_C \cup \mathcal{T}_E$ can be rewritten (according to $\mathcal{T}_D$) to a finite set of BCQs, denoted by $\gamma(\mathcal{T}_C \cup \mathcal{T}_E, \mathcal{T}_D)$, such that $\mathcal{T} \cup \mathcal{A} \models \bot$ if and only if $\mathcal{A} \models Q'$ for some $Q' \in \gamma(\mathcal{T}_C \cup \mathcal{T}_E, \mathcal{T}_D)$;

(3) Q can be rewritten (according to $\mathcal{T}_D$) to a finite set of BCQs, denoted by $\tau(Q, \mathcal{T}_D)$, such that $\mathcal{T}_D \cup \mathcal{A} \models Q$ if and only if $\mathcal{A} \models Q'$ for some $Q' \in \tau(Q, \mathcal{T}_D)$.

A TBox will be first-order rewritable if it can be translated to a Datalog$^\pm$ ontology that consists of linear (multilinear, sticky, or sticky-join) TGDs, constraints and special EGDs that are separable from TGDs (Calì, Gottlob, and Lukasiewicz 2012; Calì, Gottlob, and Pieris 2012). For many DLs in the DL-Lite family (Calvanese et al. 2007), such translations exist and have been given in (Calì, Gottlob, and Lukasiewicz 2012; Calì, Gottlob, and Pieris 2012). Since the DL-Lite family has become popular in many ontology applications, such as the OBDA systems, first-order rewritable TBoxes are sufficient for these applications. Therefore, we focus on first-order rewritable TBoxes.

The Query Abduction Problem

We consider a general problem of ABox abduction, called the *query abduction problem*, which is derived from (Calvanese et al. 2013) and is defined below.

Definition 2. Let $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ be a consistent DL ontology, Q be a BCQ and Σ be a set of concept names and role names (called *abducible predicates*). We call $\mathcal{P} = (\mathcal{T}, \mathcal{A}, Q, \Sigma)$ an instance of the *query abduction problem*. An *explanation* for $\mathcal{P}$ is a Σ-ABox $\mathcal{E}$ such that $\mathcal{T} \cup \mathcal{A} \cup \mathcal{E} \models Q$ and $\mathcal{T} \cup \mathcal{A} \cup \mathcal{E} \not\models \bot$. The set of explanations for $\mathcal{P}$ is denoted by $\mathsf{expl}(\mathcal{P})$.

A traditional task for ABox abduction (Klarman, Endriss, and Schlobach 2011; Du et al. 2011a) requires to compute all explanations with certain minimality. Since an explanation defined above may contain fresh individuals not in $\mathcal{P}$ (i.e. neither in $\mathcal{O}$ nor in Q), we cannot use standard set-inclusion minimality as in (Du et al. 2011a). To compare with explanations containing different fresh individuals, we treat fresh individuals as variables. Analogously, we define a *substitution* for an explanation $\mathcal{E}$ as a mapping from fresh individuals in $\mathcal{E}$ to existing or fresh individuals, and a *renaming* for $\mathcal{E}$ as a substitution for $\mathcal{E}$ that maps different fresh individuals to different fresh individuals. Below we introduce a variant set-inclusion relation and the notion of *minimal explanation*.

Definition 3. For two explanations $\mathcal{E}$ and $\mathcal{E}'$ for $\mathcal{P} = (\mathcal{T}, \mathcal{A}, Q, \Sigma)$, by $\mathcal{E}' \subset_r \mathcal{E}$ we denote that there exists a renaming ρ for $\mathcal{E}'$ such that $\mathcal{E}'\rho \subset \mathcal{E}$. A *minimal explanation* $\mathcal{E}$ for $\mathcal{P}$ is an explanation for $\mathcal{P}$ such that there is no explanation $\mathcal{E}'$ for $\mathcal{P}$ fulfilling $\mathcal{E}' \subset_r \mathcal{E}$. The set of different minimal explanations for $\mathcal{P}$ up to renaming of fresh individuals (simply up to renaming) is denoted by $\mathsf{mexpl}(\mathcal{P})$.

As discussed in the first section, $\mathsf{mexpl}(\mathcal{P})$ can have infinitely many explanations for $\mathcal{P}$. However, when $\mathcal{T}$ is restricted to be first-order rewritable, $\mathsf{mexpl}(\mathcal{P})$ will become a finite set. This conclusion is drawn from Lemma 1, where a *bipartition* of a BCQ Q is a tuple of two BCQs (Q_1, Q_2) such that $Q_1 \cap Q_2 = \emptyset$ and $Q_1 \cup Q_2 = Q$.

Lemma 1. *For every minimal explanation $\mathcal{E}$ for $\mathcal{P} = (\mathcal{T}, \mathcal{A}, Q, \Sigma)$, there exists a BCQ $Q' \in \tau(Q, \mathcal{T}_D)$, a* bipartition *$(Q_1, Q_2)$ of Q', a ground substitution θ for Q_2 and a ground substitution σ for $Q_1\theta$ such that $\mathcal{E} = Q_1\theta\sigma$, $Q_2\theta \subseteq \mathcal{A}$, $\mathcal{T} \cup \mathcal{A} \cup Q_1\theta\sigma \not\models \bot$.*

Let $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ denote the set $\{Q_1\theta \mid Q' \in \tau(Q, \mathcal{T}_D)$, (Q_1, Q_2) is a bipartition of Q', θ is a ground substitution for Q_2 such that Q_1 contains only predicates in Σ, $Q_2\theta \subseteq \mathcal{A}$, $\mathcal{T} \cup \mathcal{A} \cup Q_1\theta \not\models \bot\}$. Lemma 1 shows that $\mathsf{mexpl}(\mathcal{P})$ is a subset of $\{E\sigma \mid E \in \Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$, σ is a ground substitution for $E\}$ up to renaming. Since the number of BCQs in $\tau(Q, \mathcal{T}_D)$, the number of bipartitions of a BCQ in $\tau(Q, \mathcal{T}_D)$ and the number of different explanations of the form $E\sigma$ for an element $E \in \Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ are finite while $\tau(Q, \mathcal{T}_D)$ is independent from $\mathcal{A}$, the cardinality of $\mathsf{mexpl}(\mathcal{P})$ is finite and is only polynomial in the cardinality of $\mathcal{A}$ under an assumption that the size of $\mathcal{T}$ and the size of Q are constants. Moreover, by Condition (2) in Definition 1, $\mathcal{T} \cup \mathcal{A} \cup Q_1\theta \not\models \bot$ if and only if there is no $Q' \in \gamma(\mathcal{T}_C \cup \mathcal{T}_E, \mathcal{T}_D)$ such that $\mathcal{A} \cup Q_1\theta \not\models Q'$. Hence the consistency checking for $\mathcal{T} \cup \mathcal{A} \cup Q_1\theta$ can be done in polynomial time in data complexity, and so can $\mathsf{mexpl}(\mathcal{P})$ be computed. However, $\mathsf{mexpl}(\mathcal{P})$ can still be too large to be computed, as shown in the first section. To reduce the number of explanations that need to be computed, below we introduce a subsumption relation among explanations and the notion of *representative explanation*.

Definition 4. A explanation $\mathcal{E}$ for $\mathcal{P}$ is said to be *subsumed by* another explanation $\mathcal{E}'$ for $\mathcal{P}$, denoted by $\mathcal{E}' \subseteq_s \mathcal{E}$, if there is a substitution θ for $\mathcal{E}'$ such that $\mathcal{E}'\theta \subseteq \mathcal{E}$; it is said to be *strictly subsumed by* $\mathcal{E}'$, denoted by $\mathcal{E}' \subset_s \mathcal{E}$, if $\mathcal{E}' \subseteq_s \mathcal{E}$ and $\mathcal{E} \not\subseteq_s \mathcal{E}'$. A *representative explanation* $\mathcal{E}$ for $\mathcal{P}$ is a minimal explanation for $\mathcal{P}$ such that there is no minimal explanation $\mathcal{E}'$ for $\mathcal{P}$ fulfilling $\mathcal{E}' \subset_s \mathcal{E}$. The set of different representative explanations for $\mathcal{P}$ up to renaming is denoted by $\mathsf{rexpl}(\mathcal{P})$.

An example for minimal explanations and representative explanations is given below.

Example 1. Let $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ be a DL ontology. The TBox $\mathcal{T}$ consists of the following three axioms:
$\alpha_1 : \mathsf{Student} \sqsubseteq \mathsf{Person}$ $\quad \alpha_2 : \mathsf{Student} \sqsubseteq \neg\mathsf{Employee}$
$\alpha_3 : \exists\mathsf{worksFor}.\top \sqsubseteq \mathsf{Employee}$
The ABox $\mathcal{A}$ consists of the following two assertions:
$\alpha_4 : \mathsf{Person}(\mathsf{Tom})$ $\quad \alpha_5 : \mathsf{Student}(\mathsf{Amy})$
Suppose the set Σ of abducible predicates is $\{\mathsf{Student}, \mathsf{Employee}, \mathsf{worksFor}\}$. For the BCQ $Q_1 = \{\mathsf{Person}(\mathsf{Tom}), \mathsf{worksFor}(\mathsf{Tom}, x)\}$, there are three minimal explanations for $(\mathcal{T}, \mathcal{A}, Q_1, \Sigma)$, namely $\{\mathsf{worksFor}(\mathsf{Tom}, \mathsf{Tom})\}$, $\{\mathsf{worksFor}(\mathsf{Tom}, \mathsf{Amy})\}$ and $\{\mathsf{worksFor}(\mathsf{Tom}, u)\}$ where u denotes a fresh individual, among which the last one is the unique representative explanation for $(\mathcal{T}, \mathcal{A}, Q_1, \Sigma)$. For the BCQ $Q_2 = \{\mathsf{Person}(\mathsf{Amy}), \mathsf{worksFor}(\mathsf{Amy}, x)\}$, there is no minimal explanation for $(\mathcal{T}, \mathcal{A}, Q_2, \Sigma)$, because a minimal explanation should contain an assertion of the form $\mathsf{worksFor}(\mathsf{Amy}, a)$ which is inconsistent with $\{\alpha_2, \alpha_3, \alpha_5\}$.

We propose to compute $\mathsf{rexpl}(\mathcal{P})$ instead of $\mathsf{mexpl}(\mathcal{P})$ because the cardinality of $\mathsf{rexpl}(\mathcal{P})$ can be much smaller than that of $\mathsf{mexpl}(\mathcal{P})$, and as shown in Theorem 1, $\mathsf{mexpl}(\mathcal{P})$

can be retrieved from $\mathsf{rexpl}(\mathcal{P})$ by substituting fresh individuals, performing consistency checks and $\subset_r$-checks, and by deleting duplicate explanations up to renaming.

Theorem 1. *For a set S of explanations for $\mathcal{P}$, let $\mathsf{reduce}_r(S)$ denote the set of explanations obtained from $\{\mathcal{E} \in S \mid$ there is no $\mathcal{E}' \in S$ such that $\mathcal{E}' \subset_r \mathcal{E}\}$ by deleting all duplicate explanations up to renaming. We have $\mathsf{mexpl}(\mathcal{P}) = \mathsf{reduce}_r(\{\mathcal{E}\theta \mid \mathcal{E} \in \mathsf{rexpl}(\mathcal{P}), \theta$ is a substitution for $\mathcal{E}$ such that $\mathcal{T} \cup \mathcal{A} \cup \mathcal{E}\theta \not\models \bot\})$ up to renaming.*

Computing all Representative Explanations

Given $\mathcal{P} = (\mathcal{T}, \mathcal{A}, Q, \Sigma)$, the explanations for $\mathcal{P}$ can be computed from the aforementioned set $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q) = \{Q_1\theta \mid Q' \in \tau(Q, \mathcal{T}_D)$, (Q_1, Q_2) is a bipartition of Q', θ is a ground substitution for Q_2 such that Q_1 contains only predicates in Σ, $Q_2\theta \subseteq \mathcal{A}$, $\mathcal{T} \cup \mathcal{A} \cup Q_1\theta \not\models \bot\}$ by applying ground substitutions, as shown in the following lemma.

Lemma 2. *Let E be an element in $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ and σ a ground substitution for E, then $E\sigma$ is explanation for $\mathcal{P}$ if and only if $\mathcal{T} \cup \mathcal{A} \cup E\sigma \not\models \bot$.*

However, to efficiently compute all representative explanations for $\mathcal{P}$, it is unwise to enumerate all ground substitutions for elements in $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$. Actually, we can only consider a small portion of ground substitutions for elements in $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$. We call a ground substitution σ for a BCQ Q a *fresh substitution* for Q in $\mathcal{P}$ if it only maps variables in Q to fresh individuals not in $\mathcal{P}$.

By $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ we denote the set obtained from $\{E\sigma \mid E \in \Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$, σ is a fresh substitution for E in $\mathcal{P}$ such that $\mathcal{T} \cup \mathcal{A} \cup E\sigma \not\models \bot\}$ by deleting all duplicate elements up to renaming. By Lemma 2, all elements in $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ are explanations for $\mathcal{P}$. The following lemma shows that all representative explanations for $\mathcal{P}$ can be found in $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$.

Lemma 3. *For an arbitrary $\mathcal{E} \in \mathsf{rexpl}(\mathcal{P})$, there exists a renaming ρ for $\mathcal{E}$ such that $\mathcal{E}\rho \in \Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$.*

The set $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ may contain explanations for $\mathcal{P}$ that are not representative. The following two lemmas show that the set of representative explanations for $\mathcal{P}$ can be obtained from $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ by dropping non-minimal explanations and non-representative explanations in turn, where the $\subset_r$-checks and the $\subset_s$-checks are only applied to explanations in $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$.

Lemma 4. *For any $\mathcal{E} \in \Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$, $\mathcal{E}$ is a minimal explanation for $\mathcal{P}$ if and only if there is no $\mathcal{E}' \in \Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ such that $\mathcal{E}' \subset_r \mathcal{E}$.*

Lemma 5. *For any $\mathcal{E} \in \mathsf{reduce}_r(\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q))$ where $\mathsf{reduce}_r(S)$ is defined in Theorem 1, $\mathcal{E}$ is a representative explanation for $\mathcal{P}$ if and only if there is no $\mathcal{E}' \in \mathsf{reduce}_r(\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q))$ such that $\mathcal{E}' \subset_s \mathcal{E}$.*

By Lemma 3, Lemma 4 and Lemma 5, we obtain a direct method for computing $\mathsf{rexpl}(\mathcal{P})$ without computing $\mathsf{mexpl}(\mathcal{P})$ beforehand. The method together with its soundness and completeness are formally shown in the following theorem.

Theorem 2. *For a set S of explanations for $\mathcal{P}$, let $\mathsf{reduce}_s(S)$ denote the set of explanations obtained from $\{\mathcal{E} \in S \mid$ there is no $\mathcal{E}' \in S$ such that $\mathcal{E}' \subset_s \mathcal{E}\}$ by deleting all duplicate explanations up to renaming. We have $\mathsf{rexpl}(\mathcal{P}) = \mathsf{reduce}_s(\mathsf{reduce}_r(\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)))$ up to renaming.*

It should be mentioned that the intermediate step, namely dropping non-minimal explanations from $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$, is crucial in guaranteeing the soundness of the above method. We show this by the following example.

Example 2. Let $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ be a DL ontology, where the TBox $\mathcal{T}$ consists of the following two axioms α_1 and α_2, and the ABox $\mathcal{A}$ has only the following assertion α_3.

$\alpha_1 : \mathsf{Student} \sqcap \exists \mathsf{hasJob}.\top \sqsubseteq \mathsf{Parttime}$

$\alpha_2 : \exists \mathsf{hasJob}.\top \sqsubseteq \mathsf{Worker} \quad \alpha_3 : \mathsf{Student}(\mathsf{Tom})$

Suppose the set Σ of abducible predicates is $\{\mathsf{hasJob}\}$. Consider the problem of computing all representative explanations for $\mathcal{P} = (\mathcal{T}, \mathcal{A}, Q, \Sigma)$, where $Q = \{\mathsf{Parttime}(\mathsf{Tom}), \mathsf{Worker}(\mathsf{Tom})\}$. It is clear that $\mathcal{T}_D = \mathcal{T}$ and $\mathcal{T}_C = \mathcal{T}_E = \emptyset$. We assume that $\tau(Q, \mathcal{T}_D) = \{Q'\}$ where $Q' = \{\mathsf{Student}(\mathsf{Tom}), \mathsf{hasJob}(\mathsf{Tom}, x), \mathsf{hasJob}(\mathsf{Tom}, y)\}$. This assumption is reasonable since for an arbitrary ABox $\mathcal{A}'$, we have $\mathcal{T}_D \cup \mathcal{A}' \models Q$ if and only if $\mathcal{A}' \models Q'$. Then $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ has a single element $\{\mathsf{hasJob}(\mathsf{Tom}, x), \mathsf{hasJob}(\mathsf{Tom}, y)\}$. By applying fresh substitutions for this element, we obtain two explanations $\mathcal{E} = \{\mathsf{hasJob}(\mathsf{Tom}, u)\}$ and $\mathcal{E}' = \{\mathsf{hasJob}(\mathsf{Tom}, u_1), \mathsf{hasJob}(\mathsf{Tom}, u_2)\}$ in $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$. It can be seen that $\mathcal{E} \subseteq_s \mathcal{E}'$, $\mathcal{E}' \subseteq_s \mathcal{E}$ and $\mathcal{E} \subset_r \mathcal{E}'$. Hence, if we do not drop non-minimal explanations from $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$, we cannot prune $\mathcal{E}'$ which is not a minimal explanation for $\mathcal{P}$.

Consider the time complexity for the above method. As analyzed in the previous section, $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ has polynomially many elements and can be computed in polynomial time in terms of data complexity. By Condition (2) in Definition 1, the consistency checks performed in the course of computing $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ from $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ can be done in polynomial time too. Hence $\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ can be computed in polynomial time in terms of data complexity, and so can $\mathsf{reduce}_s(\mathsf{reduce}_r(\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)))$ be computed.

The above method can be efficiently implemented by a level-wise strategy, where the level number k is from 0 to the maximum cardinality of BCQs in $\tau(Q, \mathcal{T}_D)$, and in the k^{th} level only elements in $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ whose cardinality equals to k are considered. This strategy can often be highly effective in pruning explanations that are not minimal or not representative. An example is given below for illustrating the above method with the level-wise strategy.

Example 3. Consider again $\mathcal{O}$ and $\mathcal{P} = (\mathcal{T}, \mathcal{A}, Q_1, \Sigma)$ that are given in Example 1. It is not hard to see that $\mathcal{T}$ is a first-order rewritable TBox such that $\mathcal{T}_D = \{\alpha_1, \alpha_2\}$, $\mathcal{T}_C = \{\alpha_3\}$, $\mathcal{T}_E = \emptyset$, $\gamma(\mathcal{T}_C \cup \mathcal{T}_E, \mathcal{T}_D) = \{Q'\}$ and $\tau(Q_1, \mathcal{T}_D) = \{Q'_1, Q'_2\}$, where $Q' = \{\mathsf{Student}(x), \mathsf{Employee}(x)\}$, $Q'_1 = \{\mathsf{Person}(\mathsf{Tom}), \mathsf{worksFor}(\mathsf{Tom}, x)\}$ and $Q'_2 = \{\mathsf{Student}(\mathsf{Tom}), \mathsf{worksFor}(\mathsf{Tom}, x)\}$. We have $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q) = \{\{\mathsf{worksFor}(\mathsf{Tom}, x)\}, \{\mathsf{Student}(\mathsf{Tom}), \mathsf{worksFor}(\mathsf{Tom}, x)\}\}$. Let S be a set storing representative explanations for $\mathcal{P}$. Initially, S is set as empty. We cope with every element in $\Xi_{\mathcal{T},\mathcal{A},\Sigma}(Q)$ in a level-wise manner, where the level number is from 0 to 2. In Level 0, there is no element to be handled. In Level 1, we handle

Table 1: The characteristics of test ontologies

Ontology	#C	#R	#TA	#AA	#I
Semintec	60	16	203	65,240	17,941
Vicodi	194	12	223	116,181	33,238
LUBM1$\sim$100	43	32	88	100,543$\sim$ 13,824,437	17,174$\sim$ 2,179,766

Note: #C/#R/#TA/#AA/#I is the number of concept names/ role names/TBox axioms/ABox assertions/individuals.

$\{\mathsf{worksFor}(\mathsf{Tom}, x)\}$. By applying fresh substitutions for it, we get $\mathcal{E}_1 = \{\mathsf{worksFor}(\mathsf{Tom}, u)\}$, where u denotes a fresh individual. Since $\mathcal{A} \cup \mathcal{E}_1 \not\models Q'$, we have $\mathcal{T} \cup \mathcal{A} \cup \mathcal{E}_1 \not\models \perp$ and thus $\mathcal{E}_1$ can be in $\mathsf{reduce}_s(\mathsf{reduce}_r(\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)))$. We add $\mathcal{E}_1$ to S. In Level 2, we handle $\{\mathsf{Student}(\mathsf{Tom}), \mathsf{worksFor}(\mathsf{Tom}, x)\}$. By applying fresh substitutions for it, we get $\mathcal{E}_2 = \{\mathsf{Student}(\mathsf{Tom}), \mathsf{worksFor}(\mathsf{Tom}, u)\}$. Since $\mathcal{E}_1 \subset_r \mathcal{E}_2$, $\mathcal{E}_2$ cannot be in $\mathsf{reduce}_s(\mathsf{reduce}_r(\Gamma_{\mathcal{T},\mathcal{A},\Sigma}(Q)))$. That is, $\mathcal{E}_2$ is pruned according to $\mathcal{E}_1$ without checking the consistency of $\mathcal{T} \cup \mathcal{A} \cup \mathcal{E}_2$. Finally, we obtain $\mathsf{rexpl}(\mathcal{P}) = S = \{\mathcal{E}_1\}$.

Experimental Evaluation

The proposed method was implemented in Java, using the Requiem (Pérez-Urbina, Motik, and Horrocks 2010) API for query rewriting and the MySQL engine to store and access ABoxes. Seven benchmark ontologies with large ABoxes were used. The first two are Semintec (about financial services) and Vicodi (about European history). The remaining ontologies are LUBMn ($n = 1, 5, 10, 50, 100$) from the Lehigh University Benchmark (Guo, Pan, and Heflin 2005), where n is the number of universities. These ontologies have TBoxes that are almost first-order rewritable and have been used to compare different DL reasoners (Motik and Sattler 2006) and to verify methods for ABox abduction (Du et al. 2011a). We removed some TBox axioms that Requiem cannot handle from the above ontologies, making the remaining portions of TBoxes first-order rewritable. The characteristics of all test ontologies are reported in Table 1. All experiments were conducted on a laptop with Intel Dual-Core 2.20GHz CPU and 4GB RAM, running Windows 7, where the maximum Java heap size was set to 1GB.

The experiments are divided into two parts.

In the first part, we compared our proposed method with the Prolog-based method[1] proposed in (Du et al. 2011a) on atomic BCQs that consist of single assertions. The Prolog-based method is the state-of-the-art method for computing all minimal explanations. Basically, it transforms a DL ontology to a Prolog program and applies a Prolog engine to compute explanations. For this part, we randomly generated thirty concept assertions as observations for each test ontology. In particular, for all LUBMn ontologies we used the same set of observations so as to verify the scalability against the increasing number n of universities. Each generated observation is not entailed by the test ontology, nor is the negation of it. Both methods work in two phases, namely the preprocessing phase and the query phase. In the preprocessing phase, the proposed method loads the TBox $\mathcal{T}$ and computes $\gamma(\mathcal{T}_C \cup \mathcal{T}_E, \mathcal{T}_D)$, while the Prolog-based method transforms the ontology to a Prolog program and loads it to the Prolog engine. In the query phase, both methods handle all observations one by one.

The proposed method always finishes the preprocessing phase in one second for all test ontologies. In contrast, the Prolog-based method spends twenty seconds to several hours in the preprocessing phase; it even runs out of memory when loading the transformed Prolog programs to the Prolog engine for LUBM10$\sim$100.

We focused more on the query phase and set a one-hour time limit to both methods for handling one observation. We first set all concept names but no role names as abducible predicates. The comparison results on the query phase are reported in Table 2. For all cases where the Prolog-based method does not run out of memory, the set of representative explanations computed by the proposed method is the same as the set of minimal explanations computed by the Prolog-based method. Moreover, the proposed method is much more efficient and more scalable against the increasing number n of universities for LUBMn ontologies. The Prolog-based method is relatively inefficient, because it cannot guarantee to compute a minimal explanation in polynomial time in data complexity. We then set all concept names and all role names as abducible predicates. The superiority of the proposed method is more evident. The Prolog-based method only handles 27 observations for Vicodi, none of observations for Semintec and three observations for LUBM1 and LUBM5, without exceeding one hour for each observation. It works so badly because there are often many minimal explanations for an observation when role names are used as abducible predicates. In contrast, the proposed method handles every observation in 1.5 seconds (most in milliseconds) except for one observation which is over the concept name $\mathsf{Student}$ for each LUBMn ontology. It spends much time on the $\mathsf{Student}$ case because the number of representative explanations is large in this case: the proposed method computes 55,491 representative explanations in 15 minutes on LUBM1 and exceeds one hour on LUBM5$\sim$100.

In the second part of experiments, we verified how well the proposed method works for general BCQs that may contain existentially quantified variables. We cannot compare it with the Prolog-based method since the Prolog-based method does not fully support general BCQs. We randomly generated twenty BCQs as observations for each of the 14 benchmark conjunctive queries (CQs) company with LUBM (Guo, Pan, and Heflin 2005). Each BCQ is not entailed by any LUBMn ontology and was generated from a benchmark CQ by replacing the first variable with an individual and keeping other variables as existentially quantified variables. We tested these BCQs on LUBMn ontologies with all concept names and all role names set as abducible predicates. Let Gn denote the group of BCQs generated from the n^{th} benchmark CQ. Figure 1 shows the average execution time for handling a BCQ in each group against the increasing number n of universities. In the figure, part (a) shows the results about relatively easy groups in which BCQs can

[1] http://dataminingcenter.net/abduction/

Table 2: The comparison results when only concept names are used as abducible predicates

	The Proposed Method			Prolog-based Method		
Ontology	avg.t	max.t	avg.#re	avg.t	max.t	avg.#me
Semintec	21	72	4.0	1,808	2,650	4.0
Vicodi	9	34	6.3	5,211	21,580	6.3
LUBM1	44	403	2.4	901	5,570	2.4
LUBM5	43	375	2.4	8,412	52,438	2.4
LUBM10	44	371	2.4	running out of memory		
LUBM50	44	388	2.4	running out of memory		
LUBM100	45	382	2.4	running out of memory		

Note: avg.t/max.t is the average/maximum execution time (in milliseconds) for handling an observation; avg.#me/avg.#re is the average number of minimal explanations/representative explanations for an observation.

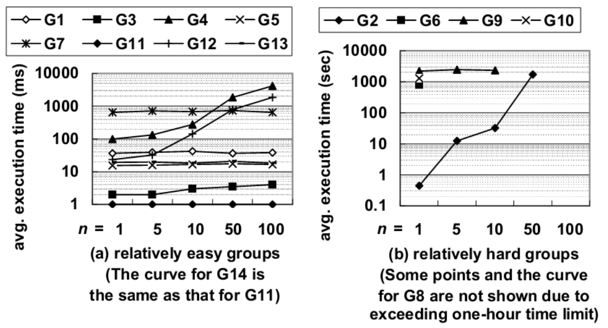

Figure 1: The results for handling general BCQs in LUBMn

be handled in four seconds for any LUBMn ontology, while part (b) shows the results about other groups where the execution time exceeding one hour is not shown. Nine out of the 14 groups are relatively easy. Especially, one BCQ in G3, G11 and G14 can be handled in a few milliseconds on average. All BCQs in G7 and all relatively hard groups except G2 contain atoms over Student. These atoms have been shown to cause many representative explanations in the first part of experiments. The BCQs in G2 have also a number of representative explanations. In more detail, the average number of representative explanations for a BCQ in G2 is from 1,007 (on LUBM1) to 67,440 (on LUBM50). These results show that the proposed method can also efficiently compute representative explanations for general BCQs.

Related Work

Abductive reasoning is a new promising research area in the context of DLs (Elsenbroich, Kutz, and Sattler 2006), where three kinds of abductive problems have been studied. Besides ABox abduction, the other two kinds are concept abduction and TBox abduction. Concept abduction computes all concepts that are added as conjunctive components to a given satisfiable concept to make the concept subsumed by an observation (which is also a concept) and remain satisfiable. Some tableau-based methods for concept abduction have been proposed in (Noia, Sciascio, and Donini 2007; 2009), while the complexity for concept abduction has been studied in (Bienvenu 2008). TBox abduction computes all sets of abducible axioms that are appended to a TBox to enforce the entailment of an observation which is a concept inclusion axiom. An automata-based method for TBox abduction has been proposed in (Hubauer, Lamparter, and Pirker 2010). The methods for concept abduction or TBox abduction are not specifically designed for ABox abduction. There is no empirical evidence that these methods can be practically adapted to ABox abduction and are able to handle a large number of ABox assertions.

As mentioned in the first section, two methods for ABox abduction have been proposed in (Klarman, Endriss, and Schlobach 2011) and in (Du et al. 2011a), respectively. The latter has also been extended to handle the cases where arbitrary concepts are used as abducible predicates (Du et al. 2012). The query abduction problem is a general problem of ABox abduction and is formally defined in (Calvanese et al. 2013), where the complexity for DL-Lite$_{\mathcal{A}}$ (a widely used DL in the DL-Lite family) is systematically studied but no method for computing all (minimal) explanations is given. As shown in (Du et al. 2011b), the method proposed in (Du et al. 2011a) can be adapted to the query abduction problem for computing some minimal explanations for a BCQ. However, it does not work in polynomial time in data complexity even for first-order rewritable TBoxes. Moreover, it is significantly less efficient and less scalable than the proposed method here as shown by our experimental results.

Conclusion and Future Work

ABox abduction is an important reasoning mechanism for DL ontologies, but there is still a lack of efficient computational methods, especially for the query abduction problem which is a general problem of ABox abduction. In this paper we have addressed the computational aspect of this problem and made the following contributions. First, considering that the number of minimal explanations for a BCQ can be infinite, we identified first-order rewritable TBoxes that guarantee the existence of finitely many minimal explanations. Second, in order to reduce the number of explanations that need to be computed, we introduced representative explanations from which minimal explanations can be retrieved. Third, we proposed a tractable method (in data complexity) for computing all representative explanations in a consistent DL ontology whose TBox is first-order rewritable. Last but not least, we empirically showed that computing the set of representative explanations is much more practical than computing the set of minimal explanations; moreover, the proposed method is efficient and scalable for DL ontologies with large ABoxes.

There are at least three directions that can be explored in the future work. Firstly, the set of representative explanations can still be large in some cases. It calls for specific treatments for these hard cases. Secondly, it is useful to develop methods for checking whether an arbitrary TBox is first-order rewritable. Lastly, besides the class of first order rewritable TBoxes, it is important to identify other classes of TBoxes or some classes of BCQs that have finitely many minimal explanations for the query abduction problem.

Acknowledgements

This work is partly supported by NSFC grants (61375056, 61005043 and 61379043), Business Intelligence Key Team of Guangdong University of Foreign Studies (TD1202), Guangdong Natural Science Foundation (S2013010012928), China National 973 project (2014CB340301) as well as ARC (DP110101042 and DP130102302).

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Bada, M.; Mungall, C.; and Hunter, L. 2008. A call for an abductive reasoning feature in owl-reasoning tools toward ontology quality control. In *Proceedings of the 5th OWLED Workshop on OWL: Experiences and Directions*.

Bienvenu, M. 2008. Complexity of abduction in the $\mathcal{EL}$ family of lightweight description logics.

Borgida, A.; Calvanese, D.; and Rodriguez-Muro, M. 2008. Explanation in the DL-Lite family of description logics. In *Proceedings of the 7nd International Conference On Ontologies, DataBases, and Applications of Semantics (ODBASE)*, 1440–1457.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A general datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics* 14:57–83.

Calì, A.; Gottlob, G.; and Pieris, A. 2012. Towards more expressive ontology languages: The query answering problem. *Artificial Intelligence* 193:87–128.

Calvanese, D.; Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning* 39(3):385–429.

Calvanese, D.; Ortiz, M.; Simkus, M.; and Stefanoni, G. 2013. Reasoning about explanations for negative query answers in DL-Lite. *Journal of Artificial Intelligence Research* 48:635–669.

Du, J.; Qi, G.; Shen, Y.; and Pan, J. Z. 2011a. Towards practical ABox abduction in large OWL DL ontologies. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, 1160–1165.

Du, J.; Wang, S.; Qi, G.; Pan, J. Z.; and Hu, Y. 2011b. A new matchmaking approach based on abductive conjunctive query answering. In *Proceedings of the 1st Joint International Semantic Technology Conference (JIST)*, 144–159.

Du, J.; Qi, G.; Shen, Y.; and Pan, J. Z. 2012. Towards practical ABox abduction in large description logic ontologies. *International Journal on Semantic Web and Information Systems* 8(2):1–33.

Du, J.; Wang, K.; and Shen, Y. 2014. A tractable approach to ABox abduction over description logic ontologies. Technical report, Guangdong University of Foreign Studies. http://www.dataminingcenter.net/abduction/AAAI14-TR.pdf.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42(1):3–42.

Elsenbroich, C.; Kutz, O.; and Sattler, U. 2006. A case for abductive reasoning over ontologies. In *Proceedings of the 3rd OWLED Workshop on OWL: Experiences and Directions*.

Guo, Y.; Pan, Z.; and Heflin, J. 2005. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics* 3(2–3):158–182.

Hubauer, T.; Lamparter, S.; and Pirker, M. 2010. Automata-based abduction for tractable diagnosis. In *Proceedings of the 23rd International Workshop on Description Logics*.

Klarman, S.; Endriss, U.; and Schlobach, S. 2011. ABox abduction in the description logic $\mathcal{ALC}$. *Journal of Automated Reasoning* 46(1):43–80.

Motik, B., and Sattler, U. 2006. A comparison of reasoning techniques for querying large description logic aboxes. In *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, 227–241.

Noia, T. D.; Sciascio, E. D.; and Donini, F. M. 2007. Semantic matchmaking as non-monotonic reasoning: A description logic approach. *Journal of Artificial Intelligence Research* 29:269–307.

Noia, T. D.; Sciascio, E. D.; and Donini, F. M. 2009. A tableaux-based calculus for abduction in expressive description logics: Preliminary results. In *Proceedings of the 22nd International Workshop on Description Logics*.

Pérez-Urbina, H.; Motik, B.; and Horrocks, I. 2010. Tractable query answering and rewriting under description logic constraints. *Journal of Applied Logic* 8(2):186–209.