Querying DL-lite Knowledge Bases from Hidden Datasets[*]

Ghassen Hamdi
MARS Research Laboratory
Univ Sousse, Tunisia
hamdighassan@gmail.com

Mohamed Nazih Omri
MARS Research Laboratory
Univ Sousse, Tunisia
mohamednazih.omri@fsm.rnu.tn

Odile Papini
LSIS, Univ Aix-Marseille, France
odile.papini@amu-univ.fr

Salem Benferhat
CRIL, Univ Artois, France
benferhat@cril.fr

Zied Bouraoui
CRIL, Univ Artois, France
bouraoui@cril.fr

Abstract

One of the main aims of Ontology-Based Data Access is to uniform querying data using knowledge encoded in an ontology. Data are often provided by several information sources, and this has led to a number of methods that merge them in order to get a unified point of view. Existing merging approaches relie on the fact that the content of the datasets is known. However in many applications such as in recommendation sites, this prior knowledge about the content of the datasets is not available. This paper investigates the problem of querying multiple information sources without knowing the content of the datasets. We first provide several strategies to answer queries from datasets. We then study how these strategies can be compared to each other from a productivity point of view.

Introduction

Structured knowledge about entities plays an important role in many web applications. Ontologies offer a powerful framework to encode structured knowledge about the concepts and properties of a given domain. They are typically expressed using description logics (Baader et al. 2010), and stored in two parts: a TBox, which encodes generic knowledge about a domain through the semantic relationships between concepts and relations, and an ABox, which contains data, i.e. information about which entities belong (resp. related) to what concepts (resp. relations).

Recent years have witnessed an increasing interest in Ontology-Based Data Access (Maurizio 2011; Poggi et al. 2008; Muro, Kontchakov, and Zakharyaschev 2013), in which structured knowledge (i.e. stored in a TBox) is exploited when querying data (i.e. stored as an ABox). Adding ontological knowledge aims, for instance, at improving query answering, merging/integrating data from different sources, etc.

In the case, when the data are provided from different sources an important research question is: how to perform query answering in a meaningful way? In a multi sources context, one should take into account of many problems such as the dissatisfaction of the sources, missing data, redundancy, etc. These problems have led to a number of works mainly aiming at merging datasources (i.e. ABoxes) in order to get a global point of view (Benferhat et al. 2014; Wang et al. 2012; Benferhat et al. 2017). To perform merging while dealing with conflicts, existing approaches require to perform some fusion operators which allow for combining the pieces of information coming from several sources (ABoxes) in order to know the whole content of the ABoxes. Unfortunately, this is not always guaranteed.

This paper studies the problem of query answering without neither merging datasources nor knowing their contents. This might make sense in certain practical settings. Consider for example the case of recommendation sites, online booking sites where the results given to users are made by only querying and without knowing the whole contents of the sources. The assumption that the data are not accessible or hidden for privacy or security reasons, is quite common, in many Web applications, however as far as we know this problem has not already been addressed in query answering mediated by an ontology.

In this paper, we provide several strategies to answer queries from hidden datasets and we study how these strategies can be compared to each other from a productivity point of view.

Consider a simple example of a Web site managing airline companies described using a rule saying that a customer can book a flight proposed by a company (for the sake of simplicity, a flight concerns departure, destination and date). Consider the flights booking as statements. Thus, we have a dataset for each company. Let us consider q_1 ("Output the list of customers of all companies") and q_2 ("A customer that booked the same flight in different companies"). For privacy purpose, it is obvious that one may not have response for query q_1. On the other hand, one may have response to q_2, but this requires merging the responses given by datasets.

The rest of this paper is organized as follows. In the next section we recall some basic notions of Description Logics and query answering. Subsequently we propose new strategies for querying multiple hidden sources. Finally, we compare the proposed strategies from a productivity point of view and present related works before concluding.

Description Logics

Description Logics (DL) are the logical framework underlying the ontology language, OWL. DL-Lite (Alessandro et

[*] This work was supported by the european projects, H2020 Marie Skodowska-Curie Actions (MSCA) Research and Innovation Staff Exchange (Rise) : Aniage (Project Number 69—215)

al. 2009) is a family of tractable DLs specifically tailored for applications that use large amounts of data, for which query answering is the most important reasoning task. DL-Lite guarantees a low computational complexity. This fact makes DL-Lite especially well suited for Ontology-Based Data Access (OBDA). In the following, we will recall $DL - Lite_R$ logic.

DL-Lite syntax and semantics. Let N_C, N_R and N_I be three pairwise disjoint sets of atomic concepts, atomic roles and individuals respectively. Let $A \in N_C$, $P \in N_R$, three connectors '$\neg$', '$\exists$' and '$^-$' are used to define complex concepts and roles. Basic concepts (*resp.* roles) B (*resp.* R) and complex concepts (*resp.* roles) C (*resp.* E) are defined in DL-Lite as follows:

$$\begin{array}{lllllll} B \longrightarrow & A & \mid & \exists R & \quad C \longrightarrow & B & \mid & \neg B \\ R \longrightarrow & P & \mid & P^- & \quad E \longrightarrow & R & \mid & \neg R \end{array}$$

where P^- represents the inverse of P. A DL-Lite knowledge base (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is called the TBox and $\mathcal{A}$ is called the ABox. A TBox includes a finite set of inclusion axioms on concepts and on roles respectively of the form: $B \sqsubseteq C$ and $R \sqsubseteq E$. The ABox contains a finite set of assertions (facts) of the form $A(a)$ and $P(a, b)$ where $A \in N_C$, $P \in N_R$ and $a, b \in N_I$.

The semantics is given in term of interpretations. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is called domain, and $.^{\mathcal{I}}$ an interpretation function that assigns to each $a \in N_I$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to each $A \in N_C$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each $P \in N_R$ an $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The function $.^{\mathcal{I}}$ is extended in a straightforward way for complex concepts and roles, e.g $(\neg B)^{\mathcal{I}}$=$\Delta^{\mathcal{I}} \setminus B^{\mathcal{I}}$, $(P^-)^{\mathcal{I}}$=$\{(y, x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x, y) \in P^{\mathcal{I}}\}$ and $(\exists R)^{\mathcal{I}}$=$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} \mid (x, y) \in R^{\mathcal{I}}\}$. An interpretation $\mathcal{I}$ is said to be a model of a concept (*resp.* role) inclusion axiom, denoted by $\mathcal{I} \models B \sqsubseteq C$ (*resp.* $\mathcal{I} \models R \sqsubseteq E$), iff $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ (*resp.* $R^{\mathcal{I}} \subseteq E^{\mathcal{I}}$). Similarly, we say that $\mathcal{I}$ satisfies a concept (*resp.* role) assertion, denoted by $\mathcal{I} \models A(a)$ (*resp.* $I \models P(a, b)$), iff $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (*resp.* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$). An interpretation $\mathcal{I}$ is said to satisfy a KB $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$, denoted $\mathcal{I} \models \mathcal{K}$, iff $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. Such interpretation is said to be a model of $\mathcal{K}$. Lastly, a TBox $\mathcal{T}$ is said to be incoherent if there exists a concept C s.t $\forall \mathcal{I}$:$\mathcal{I} \models \mathcal{T}$, we have $C^{\mathcal{I}}$=$\emptyset$. A DL-Lite KB $\mathcal{K}$ is said to be inconsistent if it does not admit any model.

Query answering. A query is a first-order logic formula, denoted q=$\{\vec{x} \,|\, \phi(\vec{x})\}$, where $\vec{x}$=(x_1,...,x_n) are free variables, n is the arity of q and atoms of $\phi(\vec{x})$ are of the form $A(t_i)$ or $P(t_i, t_j)$ with $A \in N_C$ and $P \in N_R$ and t_i, t_j are terms, *i.e.,* constants of N_I or variables. When $\phi(\vec{x})$ is of the form $\exists \vec{y}.conj(\vec{x}, \vec{y})$ where $\vec{y}$ are bound variables called existentially quantified variables, and $conj(\vec{x}, \vec{y})$ is a conjunction of atoms of the form $A(t_i)$ or $P(t_i, t_j)$ with $A \in N_C$ and $P \in N_R$ and t_i, t_j are terms, then q is said to be a conjunctive query (CQ). When n=0, then q is called a boolean query (BQ). A BQ with no bound variables is called a ground query (GQ). Lastly, when q only contains one atom with no free variables, then it is called an instance query (IQ) (*i.e.,* instance checking).

An answer to a CQ $q(\vec{x}) \leftarrow conj(\vec{x}, \vec{y})$ over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a non empty set of tuples $\vec{s} = (s_1, \cdots, s_k) \in N_I \times \cdots \times N_I$ such that $\langle \mathcal{T}, \mathcal{A} \rangle \models q(\vec{s})$ $\langle \mathcal{T}, \mathcal{A} \rangle \models q(\vec{s})$.

An answer to a boolean query BQ $q()$ w.r.t. $\mathcal{A}$ is $true$ if $\langle \mathcal{T}, \mathcal{A} \rangle \models q()$ and false otherwise.

In this paper, we focus on conjonctive queries (CQs).

Multiple Sources Query Answering Strategies

This section investigates querying multiple ABox, without knowing their contents, using a DL-Lite ontology. We first introduce the notion of DL-Lite knowledge base with multiple assertional sets. While a standard DLs knowledge base has a single ABox, it is convenient for the definitions of inference strategies to introduce the notion of "profile" which is simply a multiset of ABoxes. We denote by $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ a knowledge base with profile where $\mathcal{T}$ is a standard DL-Lite TBox and $P = \{\mathcal{A}_1, ..., \mathcal{A}_m\}$ is an ABox profile. We assume that each ABox $A_i \in P$ is consistent with the ontology, namely $\langle \mathcal{T}, \mathcal{A}_i \rangle$ is consistent.

Example 1. *Let $Customer$, $Flight$ be two concepts and $books$ a role. Consider the following knowledge base $\mathcal{K}_P = \langle \mathcal{T}, P = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\} \rangle$ where $\mathcal{T} = \{Customer \sqsubseteq \exists books, Flight \sqsubseteq \exists books^-\}$ and. $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ are respectively the datasets corresponding to "AirFrance", "KLM" and "Transavia". The ABoxes are described as follows: $\mathcal{A}_1$ = {Customer(a), Customer(b), Customer(d), Flight(f1), Flight(f2), books(a, f1), books(b, f1), books(d, f2) }, $\mathcal{A}_2$ = {Customer(a), Customer(b), Flight(f1), books(a, f1), books(b, f1)}, $\mathcal{A}_3$ = {Customer(a), Customer(c), Customer(d), Flight(f1), Flight(f2), books(a, f1), books(c, f2), books(d, f2)}.*

This example is used as a running example in the paper in order to illustrate the different proposed strategies to query a knowledge base without knowing the content of the datasets.

Query Answering Strategies

The following definition generalizes query answering for a multiple set of ABoxes.

Definition 1. *Let $\mathcal{K}_P$= $\langle \mathcal{T}, P \rangle$ be an knowledge base with $P = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$ and $q(\vec{x})$ be a conjunctive query. An answer to the query $q(\vec{x})$ w.r.t. $\mathcal{A}_i$ is a tuple $\vec{s} = (s_1, \cdots, s_k) \in N^I \times \cdots \times N^I$ such that $\langle \mathcal{T}, \mathcal{A}_i \rangle \models q(\vec{s})$.*

Let $q(\vec{x})$ be a conjunctive query, S_i denotes the set of answers to $q(\vec{x})$ w.r.t. $\mathcal{A}_i$. Namely $S_i = \{\vec{s} \in N^I \times \cdots \times N^I : \langle \mathcal{T}, \mathcal{A}_i \rangle \models q(\vec{s})\}$. We denote by $S_P = \{S_1, ..., S_m\}$ the profile, multiset of sets of answers to a query $q(\vec{x})$ with respect to P, with S_i be the set of answers to $q(\vec{x})$ w.r.t. $\mathcal{A}_i$ for $1 \leq i \leq m$. Intuitively, S_P contains all the valid answers to a query, namely the answers that can be obtained from each ABox $\mathcal{A}_i \in P$. Of course, when there is no answer to a query $q(\vec{x})$ with respect to $\mathcal{A}_i$, S_i = $\emptyset$. However having a valid answer does not mean that all the sources are in agreement with that. For instance, even though each ABox is consistent with the TBox, it might be the case that the union of these ABoxes is inconsistent. Moreover it might be the case that the sources do not share same information. In fact, one needs to access to the data stored in the ABoxes in a meaningful way.

We introduce different strategies for query answering without knowing the content of the dataset in $\mathcal{K}_P$. One way to perform query answering is to only consider answers to the query that hold in all the sources. More formally :

Definition 2. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be a knowledge base with $P = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$. Let $q(\vec{x})$ be a conjunctive query and S_P be the profile of sets of answers ("answers profile") to $q(\vec{x})$ w.r.t. P. An (*universal*) answer to the query $q(\vec{x})$ w.r.t. $\mathcal{K}_P$ is a tuple $\vec{s}$ such that*

$$\mathcal{K}_P \models_{\forall} q(\vec{s}) \quad if \quad \forall S_i \in S_P,\ \vec{s} \in S_i.$$

Note that the notion of universal inference offers a natural way to deal with conflicting sources. It was used for instance in default reasoning (Reiter 1987), where one only accepts conclusions derived from each extension of a default theory. We denote by $S_{\forall}$ the set of answers obtained by universal inference, $S_{\forall} = \{\vec{s} : \mathcal{K}_P \models_{\forall} q(\vec{s})\}$.

Example 2. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be the DL-Lite knowledge base (KB) managing flight booking introduced in Example 1. Assume that an authorized query is a query which asks for the clients that made a booking for a flight, more formally, $q(x) = \exists x\, books(x, y)$. We get the following answers $S_P = \{S_1, S_2, S_3\}$ with $S_1 = \{a, b, d\}$ the set of answers from the ABox $\mathcal{A}_1$, $S_2 = \{a, b\}$ the set of answers from the ABox $\mathcal{A}_2$ and $S_3 = \{a, c, d\}$ the set of answers from the ABox $\mathcal{A}_3$. Clearly, $\mathcal{K}_P \models_{\forall} q(a)$ since $\forall S_i \in S_P$ $a \in S_i$ and $S_{\forall} = \{a\}$ which expresses the fact that the customer a booked a flight in all airline companies.*

In some situations, it makes sense to only looking for a possible solution of a set of constraints or preferences such as in some decision problem. As a second strategy we propose the existential inference that states that an answer is deduced from at least one ABox. More formally.

Definition 3. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be a knowledge base with $P = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$. Let $q(\vec{x})$ be a conjunctive query and S_P be the answers profile to $q(\vec{x})$ w.r.t. P. An (*existential*) answer to the query $q(\vec{x})$ w.r.t. $\mathcal{K}_P$ is a tuple $\vec{s}$ such that*

$$\mathcal{K}_P \models_{\exists} q(\vec{s}) \quad if \quad \exists S_i \in S_P,\ \vec{s} \in S_i$$

We denote by $S_{\exists}$ the set of answers obtained by existential inference, $S_{\exists} = \{\vec{s} : \mathcal{K}_P \models_{\exists} q(\vec{s})\}$.

Example 3. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be the KB from Example 2. Consider again the same query $q(x) = \exists x\, books(x, y)$. Clearly, $\mathcal{K}_P \models_{\exists} q(a)$ since $a \in S_1$ moreover, $\mathcal{K}_P \models_{\exists} q(b)$, $\mathcal{K}_P \models_{\exists} q(c)$ and $\mathcal{K}_P \models_{\exists} q(d)$, therefore $S_{\exists} = \{a, b, c, d\}$ which expresses the fact that the customers a, b, c and d booked a flight in at least one airline company.*

The existential query answering strategy is a very adventurous strategy. To this end, we propose to consider a majority-based strategy as an alternative. It states that an answer to a query is considered as valid if it can be deduced from the majority of the datasets. More formally,

Definition 4. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be a knowledge base with $P = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$. Let $q(\vec{x})$ be a conjunctive query and S_P be answers profile to $q(\vec{x})$ w.r.t. P. An (*majority*) answer to the query $q(\vec{x})$ w.r.t. $\mathcal{K}_P$ is a tuple $\vec{s}$ such that*

$$\mathcal{K}_P \models_{maj} q(\vec{s}) \quad if \quad \frac{\mid i \ : \ \vec{s} \in S_i \mid}{m} > \frac{1}{2}$$

We denote by S_{maj} the set of answers obtained by majority-based strategy, namely $S_{maj} = \{\vec{s} : \mathcal{K}_P \models_{maj} q(\vec{s})\}$. Note that this strategy offers a good compromise between universal or existential strategies.

Example 4. *Let us continue Example 2 with the same query : $q(x) = \exists x\, books(x, y)$. One can check that $\mathcal{K}_P \models_{maj} q(b)$ since $b \in S_1$ and $b \in S_2$, $\mathcal{K}_P \models_{maj} q(d)$ since $d \in S_1$ and $d \in S_3$ and $\mathcal{K}_P \models_{maj} q(a)$ since $a \in S_1$, $a \in S_2$, and $a \in S_3$, therefore $S_{maj} = \{a, b, d\}$ which expresses the fact that the customers a, b and d booked a flight in the majority of the airline companies.*

Introducing Cardinality

We now introduce new strategies obtained by adding a cardinality criterion to the strategies proposed above. We basically restrict the application of the query answering relations $\forall, \exists$ and maj to some sets of answers in the profile S_p. More precisely, we only select the largest sets in S_P with respect to cardinality . The choice of the largest sets of answers might make sense in certain practical applications. For instance, suppose that we want to know the customers that booked a flight in the most popular airline compagnies in terms of number of customers.

We denote by Sc_P the profile of sets of answers with maximal cardinality to a query $q(\vec{x})$ w.r.t. P. The set Sc_P is defined as follows:

$$Sc_P = \{S_i,\ i \in \{1, \cdots, m\} \ : \ \forall j \in \{1, \cdots, m\} \ \mid S_j \mid \leq \mid S_i \mid\}$$

The strategies presented before can be extended using cardinality criterion as follows:

Definition 5. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be a knowledge base with $P = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$. Let $q(\vec{x})$ be a conjunctive query and Sc_P be the answers profile with maximal cardinality to $q(\vec{x})$ w.r.t. P. An answer to the query $q(\vec{x})$ w.r.t. $\mathcal{K}_P$ is a tuple $\vec{s}$ such that :*

- $\mathcal{K}_P \models_{\forall_c} q(\vec{s}) \quad$ *if* $\quad \forall S_i \in Sc_P, \vec{s} \in S_i$.
- $\mathcal{K}_P \models_{\exists_c} q(\vec{s}) \quad$ *if* $\quad \exists S_i \in Sc_P, \vec{s} \in S_i$.
- $\mathcal{K}_P \models_{maj_c} q(\vec{s}) \quad$ *if* $\quad \frac{|S_i \in Sc_P \ : \ \vec{s} \in S_i|}{|Sc_P|} \geq \frac{1}{2}$.

We denote by $S_{\forall_c}$ (resp. $S_{\exists_c}$, S_{maj_c}) the set of answers obtained by universal (resp. existential, majority-based) strategies.

Example 5. *Let us continue Example 2. Clearly, $Sc_P = \{S_1, S_3\}$ then $S_{\forall_c} = \{a, d\}$ which expresses the fact that the customers a and d booked a flight in all the most popular airline companies. $Sc_P = \{S_1, S_3\}$ then $S_{\exists_c} = \{a, b, c, d\}$ which expresses the fact that the customers a, b, c and d booked a flight in at least one of the most popular airline companies.*

Example 6. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ with $P = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$ where $\mathcal{T}$ and $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, are from example 2 and the ABox $\mathcal{A}_4$ is the dataset corresponding to "TunisAir". Consider* *the same query $q(x) = \exists x\, books(x, y)$ and the set of answers to the query q from $\mathcal{A}_4$: S_4 = $\{a,\ c,\ e\}$. Clearly, $Sc_P = \{S_1, S_3, S_4\}$ and $S_{maj_c} = \{a, c, d\}$ which expresses the fact that the customers a, c and d booked a flight in the majority of the most popular airline companies.*

Productivity of Query Answering Strategies

In this section, we compare the proposed query answering strategies from a productivity point of view. We formalize this comparison as follows:

Definition 6. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be a knowledge base and let s_i and s_j be two strategies. We say that s_i is more productive than s_j, if for any ABox profile $P = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$ and any query $q(\vec{x})$, the set of answers to $q(\vec{x})$ w.r.t. P using the strategy s_j is included or equal to the set of answers to $q(\vec{x})$ w.r.t. P using the strategy s_i. (i.e., $S_{s_j} \subseteq S_{s_i}$).*

The following proposition gives an exhaustive study of productivity results for the different query answering strategies studied in this paper.

Proposition 1. *Let $\mathcal{K}_P = \langle \mathcal{T}, P \rangle$ be a knowledge base with $P = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$ and $q(\vec{x})$ a query. The comparison of the query answering strategies $\{\forall, \exists, maj, \forall_c, maj_c, \exists_c\}$, with respect to productivity, is given depicted in Figure 1.*

$$\begin{tikzcd}
\boxed{S_\forall} \arrow[r, "5"] \arrow[d, "1"'] & \boxed{S_{\forall_c}} \arrow[d, "3"] \\
\boxed{S_{maj}} \arrow[d, "2"'] & \boxed{S_{maj_c}} \arrow[d, "4"] \\
\boxed{S_\exists} & \boxed{S_{\exists_c}} \arrow[l, "6"]
\end{tikzcd}$$

Figure 1: $X \longrightarrow Y$ means that the set of answers obtained from the strategy X is also obtained from the strategy Y

Proof. We give the proof for each relation :

- (1) The proof is immediate since $\forall s \in S_\forall$, s is such that, $\forall i \in \{1, \cdots, m\}$, $s \in S_i$, thus $\frac{|i\,:\,\vec{s} \in S_i|}{m} = 1$, $s \in S_{maj}$. (Every answer given by all ABoxes, is an answer given by the majority of ABoxes).
- (2) The proof is immediate, $S_{maj} \subseteq S_\exists$. (Every answer given by the majority of ABoxes is given by one ABox). Note that by transitivity we get $S_\forall \subseteq S_\exists$.
- (3) and (4) the proofs are analogous to (1) and (2) applied to the set of answers of maximal cardinality. By transitivity, we get that $S_{\forall_c} \subseteq S_{\exists_c}$.
- (5) The proof is immediate, $S_\forall \subseteq S_{\forall_c}$, $\forall s \in S_\forall$ by definition of Sc_P, $s \in S_i$, with $S_i \in Sc_P$ thus $s \in S_{\forall_c}$.
- (6) Since $S_\exists = S_1 \cup \ldots \cup S_m$, by definition of Sc_P, we have $S_{\exists_c} \subseteq S_\exists$.

For the converse productivity relations, we provide the following counter-examples :

Example 7. *Let $\mathcal{K}_P{=}\langle \mathcal{T}, P \rangle$ be a KB such that $P{=}\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$, with $\mathcal{A}_1 = \{A(a)\}$, $\mathcal{A}_2 = \{A(a)\}$ and $\mathcal{A}_3 = \{A(b)\}$ and let $q(x) = \exists\, x\, A(x)$ be a query. The sets of answers to $q(x)$ from the ABoxes are S_1 = $\{a\}$, S_2 = $\{a\}$ and S_3 = $\{b\}$. Thus we have $S_\forall = \emptyset$, $S_{maj} = \{a\}$, $S_\exists = \{a, b\}$. We have (1) $S_\forall \subseteq S_{maj}$, (2) $S_{maj} \subseteq S_\exists$, and $S_\forall \subseteq S_\exists$. However, $S_{maj} \nsubseteq S_\forall$, $S_\exists \nsubseteq S_{maj}$ and $S_\exists \nsubseteq S_\forall$. Since the sets of answers have same cardinality we also have (3) $S_{\forall_c} \subseteq S_{maj_c}$, (4) $S_{maj_c} \subseteq S_{\exists_c}$ and $S_{\forall_c} \subseteq S_{\exists_c}$. However, $S_{maj_c} \nsubseteq S_{\forall_c}$, $S_{\exists_c} \nsubseteq S_{maj_c}$ and $S_{\exists_c} \nsubseteq S_{\forall_c}$.*

Example 8. *Let $\mathcal{K}_P{=}\langle \mathcal{T}, P \rangle$ be a KB such that $P{=}\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$, with $\mathcal{A}_1 = \{A(a),\ A(c), A(d)\}$, $\mathcal{A}_2 = \{A(a), A(c), A(b)\}$ and $\mathcal{A}_3 = \{A(c), A(e)\}$ and let $q(x) = \exists\, x\, A(x)$ be a query. The sets of answers to $q(x)$ from the ABoxes are S_1 = $\{$ a, c, d $\}$, S_2 = $\{$ a, c, b $\}$ and S_3 = $\{$ c, e $\}$. Thus we have $S_\forall = \{c\}$, $S_{maj} = \{a, c\}$, $S_\exists = \{a, b, c, d, e\}$. The sets of answers to $q(x)$ of maximal cardinality are $Sc_P = \{S_1, S_2\}$, thus $S_{\forall_c} = \{a, c\}$, $S_{\exists_c} = \{a, c, d, b\}$, $S_{maj_c} = \{a, c\}$. We have (5) $S_\forall \subseteq S_{\forall_c}$ and (6) $S_{\exists_c} \subseteq S_\exists$ however $S_{\forall_c} \nsubseteq S_\forall$ and $S_\exists \nsubseteq S_{\exists_c}$.*

□

Related Works

There are several lines of research related to our work. The first one is the knowledge bases merging which is a problem largely studied within the propositional logic setting. (Everaere, Konieczny, and Marquis 2010; Konieczny and Pino 2002; Lin and Mendelzon 1999; Revesz 1997; Baral, Kraus, and Minker 1991; Bloch et al. 2001) and recently in the setting of description logics where there are few works on merging (Benferhat, Bouraoui, and Loukil 2013; Benferhat et al. 2014; Moguillansky and Falappa 2007; Wang et al. 2012). These works aim to merge knowledge bases in order to get a global view which requires to know the whole content of the knowledge base before performing merging. Merging pieces of information need to use some fusion operators that permit to combine them while respecting different constraints between sources. Our paper differs from these works in two aspects : in the sense that it studies the problem of query answering without merging datasources neither knowing their contents.

The second one is concerned with the framework of databases querying sources that present some limitations has been investigated using views (Halevy 2001; Pottinger and Alon 2001; Pottinger and Halevy 2000), however these approaches mainly concern query optimization or rewriting and query languages.

The third research line related to our work is inconsistency-tolerant query answering which is handled in the framework of the database approaches (Marcelo, Leopoldo, and Chomicki 1999; Bertossi 2011; Chomicki 2006; Decker 2010; ten Cate, Halpert, and Kolaitis 2016), propositional logic approaches(Baral, Kraus, Minker and Subrahmanian 1992; Benferhat et al. 1993; Nebel 1994; Salem, Didier, and Henri 1997) or the description logic (Lembo et al. 2010) and recently in lightweight description logic DL-Lite (Lembo et al. 2015; Bienvenu 2012; Bienvenu and Rosati 2013; Benferhat et al. 2016; Baget, et al. 2016). These works aim to solve the problem of inconsistency by mainly computing a set of consistent subsets of assertional facts called repairs, that recover consistency with respect to the ontology, so using them to carry out query answering.

the fourth research line related to our work is querying Web Data Bases without Accessing to Data (Boughammoura et al. 2017; Boughammoura, Omri and Hlaoua 2012; Boughammoura, Hlaoua and Omri 2015). These works aim to identify Web page templates and the tag structures of a document in order to extract structured data from hidden web sources as the results returned in response to a user query.

Our work differs form that, since it has been assumed that each ABox in our knowledge base is consistent w.r.t. the TBox and the responses to queries as well. However, the fact that all the datasets are consistent with the TBox, does not mean that the union of answers to a query, which is implicitly can be seen as a set of fact, will be consistent with the TBox. Let us consider the following example.

Suppose we introduce the following axioms to the TBox of the running example (Example 2), $Person \sqsubseteq Customer, TravelAgency \sqsubseteq Customer, Person \sqsubseteq \neg TravelAgency$ and consider the query $q(x) = \exists x(book(x, y) \wedge Customer(x)$. Assume that $S_{\forall} = \{a\}$. Since each ABox is consistent w.r.t. the TBox, it might be the case that $Person(a) \in \mathcal{A}_1$ and $Person(a) \in \mathcal{A}_2$ but $TravelAgency(a) \in A_3$, therefore the answers profile is inconsistent. In order to overcome this problem, we plan to do repair when the union of the answers to a query (for instant the universal inference) is inconsistent with the knowledge base.

Conclusion

In this paper, we investigated the query answering problem when we do not know the content of the ABox (Data) in terms of inference strategies. We proposed basic inferences from the sets of answers provided by the ABoxes. We considered the cardinality of the sets of answers in defining new cardinality-based inferences. We also compared these inferences in terms of productivity.

In a future work we plan to study the logical properties of the proposed inferences, according to the Kraus Lehman and Magidor's framework (Kraus, Lehmann, and Magidor 1990). Besides, these inference strategies can be compared according to their computational complexity which is another future issue.

Finally, as mentioned in the previous Section, another direction of investigation is how to perform query answering of a knowledge base when the multiset of ABoxes is inconsistent.

References

Alessandro, A.; Diego, C.; Roman, K.; and Michael, Z. 2009. The dllite family and relations. *J Artif Int Res* 36(1):1–69.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and F. Patel-Schneider, P. eds. 2010. *The Description Logic Handbook Theory Implementation and Applications*. Cambridge University Press.

Baget, J.F.; Benferhat, S.; Bouraoui, Z.; Croitoru, M.; Mugnier, M.L.; Papini, O.; Rocher, S.; and Tabia, K. 2016. A general modifier based framework for inconsistency tolerant query answering. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning KR*.

Baral, C.; Kraus, S.; Minker, J.; and Subrahmanian, S.V. 1992. Combining knowledge bases consisting of first order theories. *Computational Intelligence* 8:45–71.

Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency management and prioritized syntax based entailment. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 640–647.

Benferhat, S.; Bouraoui, Z.; Lagrue, S.; and Rossit, J. 2014. Min based assertional merging approach for prioritized dl lite knowledge bases. 8–21.

Benferhat, S.; Bouraoui, Z.; Croitoru, M.; Papini, O.; and Tabia, K. 2016. Non objection inference for inconsistency tolerant query answering. In *IJCAI*.

Benferhat, S.; Bouraoui, Z.; Chau, M.T.; Lagrue, S.; and Rossit, J. 2017. A polynomial algorithm for merging lightweight ontologies in possibility theory under incommensurability assumption. In *International Conference on Agents and Artificial Intelligence ICAART 2017*, 415–422.

Benferhat, S.; Bouraoui, Z.; and Loukil, Z. 2013. Min based fusion of possibilistic dl-lite knowledge bases. In *Web Intelligence*, 23–28.

Bertossi, L.E. 2011. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management.

Bienvenu, M.; and Rosati, R. 2013. Tractable approximations of consistent query answering for robust ontology based data access. In *IJCAI 2013 Proceedings of the 23rd International Joint Conference on Artificial Intelligence*.

Bienvenu, M. 2012. On the complexity of consistent query answering in the presence of simple ontologies. In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence*.

Bloch, E.I.; Hunter, A.; Ayoun, A.; Benferhat, S.; Besnard, P.; Cholvy, L.; Cooke, R.; Dubois, D.; and Fargier, H. 2001. Fusion general concepts and characteristics. *International Journal of Intelligent Systems* 16:1107–1134.

C, Baral.; S, Kraus.; and J, Minker. 1991. Combining multiple knowledge bases. *IEEE Trans on Knowl and Data Eng* 3(2):208–220.

Boughammoura, R.; and Omri, M.N. 2017. Querying Deep Web Data Bases without Accessing to Data. *International Conference on Natural Computation Fuzzy Systems and Knowledge Discovery*

Boughammoura, R.; Omri, M.N.; and Hlaoua, L. 2012. Information retrieval from deep web based on visual query interpretation. *International Journal of Information Retrieval Research*, IJIRR 04, 45–59

Boughammoura, R.; Hlaoua, L.; and Omri, M.N. 2015. G-Form A Collaborative Design Approach to Regard Deep Web Form as Galaxy of Concepts. *International Conference on Cooperative Design Visualization and Engineering*, CDVE, 20–23

Chomicki, J. 2006. Consistent query answering five easy pieces. In *Proceedings of the 11th International Conference on Database Theory*, ICDT 07, 1–17. Berlin and Heidelberg: Springer Verlag.

Decker, H. 2010. Basic causes for the inconsistency tolerance of query answering and integrity checking. In *2010 Workshops on Database and Expert Systems Applications*, 318–322.

Halevy, Y. A. 2001. Answering queries using views a survey. *The VLDB Journal* 10(4):270–294.

Lin, J.; and Mendelzon, A. 1999. *Knowledge Base Merging by Majority*. Dordrecht: Springer Netherlands. 195–218.

Kraus, S.; Lehmann, D.J.; and Magidor, M. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.* 44(1-2):167–207.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D.F. 2010. Inconsistency tolerant semantics for description logics. In *Web Reasoning and Rule Systems Fourth International Conference RR 2010*, 103–117.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D.F. 2015. Inconsistency tolerant query answering in ontology based data access. *J Web Sem* 33:3–29.

Marcelo, A.; Leopoldo, B.; and Chomicki, J. 1999. Consistent query answers in inconsistent databases. In *Proceedings of the Eighteenth ACM SIGMOD SIGACT SIGART Symposium on Principles of Database Systems*, PODS 99, 68–79. New York and NY and USA: ACM.

Maurizio, L. 2011. Ontology based data management. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM 11, 5–6. New York NY USA: ACM.

Moguillansky, M.O., and Falappa, M.A. 2007. A non monotonic description logics model for merging terminologies. *Inteligencia Artificial Revista Iberoamericana de Inteligencia Artificial* 11(35):77–88.

Muro, M.R.; Kontchakov, R.; and Zakharyaschev, M. 2013. Ontology based data access ontop of databases. 558–573.

Nebel, B. 1994. Base revision operations and schemes semantics representation and complexity. In *Proc 11th European Conference on Artificial Intelligence*, 341–345. John Wiley Sons.

Everaere, P.; Konieczny, S.; and Marquis, P. 2010. Disjunctive Merging Quota and Gmin Merging Operators. *Artificial Intelligence*. To appear.

Poggi, A.; Lembo, D.; Calvanese, D.; Giacomo, G.D.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *J Data Semantics* 10:133–173.

Pottinger, R.; and Alon, H. 2001. Minicon a scalable algorithm for answering queries using views. *The VLDB Journal* 10(2-3):182–198.

Pottinger, R.; and Halevy, A. 2000. A scalable algorithm for answering queries using views. In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB00, 484–495. San Francisco and CA and USA: Morgan Kaufmann Publishers Inc.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artif Intell* 32(1):57–95.

Revesz, Z.P. 1997. On the semantics of arbitration. *International Journal of Algebra and Computation* 07(02):133–160.

Salem, B.; Didier, D.; and Henri, P. 1997. Some syntactic approaches to the handling of inconsistent knowledge bases a comparative study part 1 the flat case. *Studia Logica* 58(1):17–45.

Konieczny, S.; and Pino, P.R. 2002. On the frontier between arbitration and majority. In *Proceedings of the Eights International Conference on Principles of Knowledge Representation and Reasoning*, KR02, 109–120. San Francisco CA USA: Morgan Kaufmann Publishers Inc.

ten Cate, B.; Halpert, R.; and Kolaitis, P. 2016. Practical query answering in data exchange under inconsistency tolerant semantics. In *EDBT*, 233–244.

Wang, Z.; Wang, K.; Jin, Y.; and Qi, G. 2012. Ontomerge a system for merging dl-lite ontologies.