Query Failure Explanation in Inconsistent Knowledge Bases Using Argumentation

Abdallah Arioua, Nouredine Tamani, Madalina Croitoru, Patrice Buche

Query Failure Explanation in Inconsistent Knowledge Bases Using Argumentation

Abdallah ARIOUA [a] Nouredine TAMANI [b] Madalina CROITORU [b] and Patrice BUCHE [a]
[a] *IATE, INRA, University of Montpellier 2*
[b] *LIRMM, INRIA, University of Montpellier 2*

Abstract.
We address the problem of explaining Boolean Conjunctive Query (BCQ) failure in the presence of inconsistency within the Ontology-Based Data Access (OBDA) setting, where inconsistency is handled by the intersection of closed repairs semantics (ICR) and the ontology is represented by Datalog+/- rules. Our proposal relies on an interactive and argumentative approach where the processes of explanation takes the form of a dialogue between the *User* and the *Reasoner*. We exploit the equivalence between argumentation and ICR-semantics to prove that the *Reasoner* can always provide an answer for user's questions.

1. Introduction

In the popular ONTOLOGY-BASED DATA ACCESS setting the domain knowledge is represented by an ontology facilitating query answering over existing data [15]. In practical systems involving large amounts of data and multiple data sources, data inconsistency with respect to the ontology is unavoidable. Many inconsistency-tolerant semantics [3,2,12,13] have been proposed that rely on the notion of data repairs i.e. subsets of maximally consistent data with respect to the ontology. Query answering under these semantics may not be intuitively straightforward and can lead to loss of user's trust, satisfaction and may affect the system's usability [14]. As argued by Calvanese et al.[6] *explanation facilities* should not just account for user's "Why Q ?" question (why a query holds under a given inconsistency-tolerant semantics) but also for question like "Why not Q ?" (why a query does not hold under a given inconsistency-tolerant semantics).

The research problem addressed by this paper is the boolean conjunctive query failure explanation in inconsistent knowledge bases, precisely: "*Given an inconsistent KB and a boolean conjunctive query Q, why Q is not entailed from KB under the ICR-semantics?*". We use argumentation as an approach for explanation. We consider the logical instantiation of Dung's [10] abstract argumentation framework for OBDA in [8] and we exploit the equivalence result shown by the authors between the ICR-semantics and sceptical acceptance under preferred semantics to guarantee the existence of an explanation for any failed query. The explanation takes the form of a dialogue between the *User* and the *Reasoner* with the purpose of explaining the query failure. At each level of the dialogue, we use language-based introduced primitives such as *clarification* and

deepening to further refine the answer. The added value of our contribution lies in its significance and originality. We are the first to propose query failure explanation in the context of OBDA for **inconsistent knowledge bases** by means of **argumentation**. Our approach differs from [4,6] in handling query failure since we consider an inconsistent setting within OBDA. In addition, the work presented in [11] is neither applied to an OBDA context nor to the Datalog+/- language.

2. Background and Overview

In this section, we introduce the motivation and the context of our work and a formal definition of the addressed problem. Consider a knowledge base about university staff and students which contains inconsistent knowledge. This inconsistency is handled by ICR-semantics. The *User* might be interested in knowing why the knowledge base *does not entail* the query Q:"Luca is a student". Observe that the individual δ (e.g. Luca in the example above) is a negative answer for a conjunctive query Q (e.g. get me all the students in the example above) if and only if the boolean conjunctive query $Q(\delta)$ (e.g. student(Luca) in the example above) has failed. Hence, in this paper we concentrate only explaining the failure of a boolean conjunctive query. Let us formally introduce the problem of *Query Failure Explanation* in inconsistent knowledge bases.

Definition 1 (Query Failure Explanation Problem $\mathscr{P}$) *Let $\mathscr{K}$ be an inconsistent knowledge base, Q a Boolean Conjunctive Query such that $\mathscr{K} \not\models_{ICR} Q$. We then call $\mathscr{P} = \langle \mathscr{K}, Q \rangle$ a Query Failure Explanation Problem (QFEP).*

To address the *Query Failure Explanation Problem*, we use a logical instantiation of Dung's [10] abstract argumentation framework for OBDA in [8] ensuring that the argumentation framework used respects the rationality postulates [7].

Let us first introduce the OBDA setting and inconsistency-tolerant semantics. We consider *the positive existential* conjunctive fragment of first-order logic, denoted by FOL($\wedge$,$\exists$), which is composed of formulas built with the connectors $(\wedge, \rightarrow)$ and the quantifiers $(\exists, \forall)$. For more details about the language please check [8]. In this paper, for lack of space we simply give an example of a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ is composed of finite set of facts $\mathscr{F}$ and finite set of existential rules $\mathscr{R}$ and a finite set of negative constrains $\mathscr{N}$.

Example 1 *Let us consider an example inspired from [5]. In an enterprise, employees work in departments and use offices which are located in departments, some employees direct departments, and supervise other employees. In addition, a supervised employee cannot be a manager. A director of a given department cannot be supervised by an employee of the same department, and two employees cannot direct the same department , and an employee cannot work in more than one department. The following sets of (existential) rules $\mathscr{R}$ and negative constraints $\mathscr{N}$ model the corresponding ontology:*

$$\mathscr{R} = \begin{cases} \forall x \forall y \, (works\_in(x,y) \rightarrow emp(x)) & (r_1) \\ \forall x \forall y \, (directs(x,y) \rightarrow emp(x)) & (r_2) \\ \forall x \forall y \, (directs(x,y) \wedge works\_in(x,y) \rightarrow manager(x)) & (r_3) \\ \forall x \forall y \forall z \; locate\_office(y,z) \wedge uses\_office(x,y) \rightarrow works\_in(x,z) & (r_4) \end{cases}$$

$$\mathcal{N} = \begin{cases} \forall x \forall y\,(supervises(x,y) \wedge manager(y)) \rightarrow \bot & (n_1) \\ \forall x \forall y \forall z\,(supervises(x,y) \wedge works\_in(x,z) \wedge directs(y,z)) \rightarrow \bot & (n_2) \\ \forall x \forall y \forall z\,(works\_in(x,y) \wedge works\_in(x,z)) \rightarrow \bot & (n_3) \end{cases}$$

Let us suppose the following set of facts $\mathscr{F}$ that represent explicit knowledge:

$$\mathscr{F} = \begin{cases} directs(John,d_1) & (f_1) & directs(Tom,d_1) & (f_2) \\ directs(Tom,d_2) & (f_3) & supervises(Tom,John) & (f_4) \\ works\_in(John,d_1) & (f_5) & works\_in(Tom,d_1) & (f_6) \\ works\_in(Carlo,Statistics) & (f_7) & works\_in(Luca,Statistics) & (f_8) \\ works\_in(Jane,Statistics) & (f_9) & works\_in(Linda,Statistics) & (f_{10}) \\ uses\_office(Linda,o_1) & (f_{11}) & locate\_office(o_1,Accounting) & (f_{12}) \end{cases}$$

Let $F \subseteq \mathscr{F}$ be a set of facts and $\mathscr{R}$ be a set of rules. An $\mathscr{R}$-derivation of F in $\mathscr{K}$ is a finite sequence $\langle F_0,...,F_n \rangle$ of sets of facts s.t $F_0 = F$, and for all $i \in \{0,...,n\}$ there is a rule $r_i = (H_i, C_i) \in \mathscr{R}$ and a logical entailment from the H_i to F_i. For a set of facts $F \subseteq \mathscr{F}$ and a query Q and a set of rules $\mathscr{R}$, we say $F, \mathscr{R} \models Q$ iff there exists an $\mathscr{R}$-derivation $\langle F_0,...,F_n \rangle$ such that $F_n \models Q$. Given a set of facts $F \subseteq \mathscr{F}$ and a set of rules $\mathscr{R}$, the closure of F with respect to $\mathscr{R}$, denoted by $\mathtt{Cl}_{\mathscr{R}}(F)$ is the minimal set of all the knowledge that can be derived from a set of facts F by applying all the rules of $\mathscr{R}$. Finally, we say that a set of facts $F \subseteq \mathscr{F}$ and a set of rules $\mathscr{R}$ *entail* a fact f (and we write $F, \mathscr{R} \models f$) iff the closure of F by all the rules entails f (i.e. $\mathtt{Cl}_{\mathscr{R}}(F) \models f$).

Given a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, a set $F \subseteq \mathscr{F}$ is said to be *inconsistent* iff there exists a constraint $n \in \mathscr{N}$ such that $F \models H_n$, where H_n is the hypothesis of the constraint n. A set of facts is consistent iff it is not inconsistent. Notice that (like in classical logic) one can entail everything from an inconsistent set. A common solution [2, 12] is to construct maximal (with respect to set inclusion) consistent subsets of $\mathscr{K}$. Such subsets are called *repairs* and denoted by $\mathscr{R}epair(\mathscr{K})$. Once the repairs are computed, different semantics can be used for query answering over the knowledge base. In this paper we focus on (**I**ntersection of **C**losed **R**epairs semantics) [2] and we will denote ICR entailment as $\mathscr{K} \models_{ICR} Q$.

Example 2 *The knowledge base in Example 1 is inconsistent because the set of facts $\{f_1, f_4, f_6\} \subseteq \mathscr{F}$ is inconsistent since it violates the negative constraint n_2. To be able to reason in presence of inconsistency one has to construct first the repairs and intersect their closure. The following is of the repairs:*
$\mathscr{A}_1 = \{directs(John,d_1),\ supervises(Tom,John),\ works\_in(Linda,Statistic),$
$uses\_office(Linda,o_1),\ directs(Tom,d_1),\ directs(Tom,d_2),\ works\_in(Carlo,Statistic),$
$works\_in(Jane,Statistic),\ works\_in(Luca,Statistic),\ emp(John),\ emp(Tom),\ emp(Carlo),$
$emp(Luca),\ emp(Jane),\ emp(Linda)\}$.
The intersection of all closed repairs is:
$\bigcap_{A \in \mathscr{R}epair(\mathscr{K}))} \mathtt{Cl}_{\mathscr{R}}(A) = \{directs(Tom,d_1),\ directs(Tom,d_2),\ works\_in(Carlo,Statistics),$
$works\_in(Luca,Statistics),\ works\_in(Jane,Statistics),\ emp(Carlo),\ emp(Jane),\ emp(Luca),$
$emp(Tom),\ emp(John),\ emp(Linda)\}$.

Observe that in the intersection of all closed repairs there is $works\_in(Luca,Statistics)$. That means that $works\_in(Luca,Statistics)$ is ICR-entailed from the knowledge base. Whereas, $works\_in(Linda,Statistics)$ is not ICR-entailed since the facts about Linda are conflicting (because she works also for the department of Accounting).

3. Argumentation Framework, Deepening and Clarification

In what follows we quickly recall the definition of argumentation framework in the context of rule-based languages. We use the definition of argument of [8] and extend it to the notions of deepened and clarified arguments. Given a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, the *corresponding argumentation framework* $\mathscr{AF}_{\mathscr{K}}$ is a pair $(\texttt{Arg}, \texttt{Att})$ where $\texttt{Arg}$ is the set of arguments that can be constructed from $\mathscr{F}$ and $\texttt{Att}$ is an asymmetric binary relation called *attack* defined over $\texttt{Arg} \times \texttt{Arg}$. Given an argument a we denote by $Supp(a)$ the support of the argument and by $Conc(a)$ the conclusion.

Example 3 (Argument) *The following argument indicates that John is an employee because he directs department d_1:*
$a = (\{directs(John, d_1)\}, \{directs(John, d_1), emp(John)\}, emp(John))$.

Example 4 (Attack) *Consider the argument a of Example 3, the following argument* $b = (\{supervises(Tom, John), works\_in(Tom, d_1)\}, supervises(Tom, John) \wedge works\_in(Tom, d_1))$ *attacks a, because* $\{supervises(Tom, John) \wedge works\_in(Tom, d_1), directs(John, d_1)\}$ *is $\mathscr{R}$-inconsistent since it violates the constraint n_2.*

The results of [8] show the equivalence between sceptically acceptance under preferred semantics and ICR-entailment. Let us now propose functionalities that give the *User* the possibility to manipulate arguments to gain clarity for query answering and namely: deepening and clarification. Deepening aims at showing the reason why an argument attacks another. In our knowledge base the attack is justified by the violation of a constraint. Put differently, an argument attacks another argument if the conclusion of the former and the hypothesis of the latter are mutually exclusive. Thus deepening amounts to explain the attack between two arguments by showing the violated constraint.

Definition 2 (Deepening $\mathbb{D}$) *Given two arguments $a, b \in \mathscr{A}$. The mapping deepening denoted by $\mathbb{D}$ is a total function from the set $\texttt{Att}$ to $2^{\mathscr{N}}$ defined as follows:* $\mathbb{D}(b, a) = \{n |$
1. $n \in \mathscr{N}$ and ,
2. $\exists f \in \texttt{Supp}(a)$ such that $\texttt{Cl}_{\mathscr{R}}(\{Conc(b), f\}) \models H_n$.
$\}$
Note that H_n is the hypothesis of the constraint n.

Example 5 (Deepening) *Consider the argument a of Example 3, the argument* $b = (\{supervises(Tom, John)\ , works\_in(Tom, d_1)\}, supervises(Tom, John) \wedge works\_in(Tom, d_1))$ *attacks a, hence deepening is* $\mathbb{D}(b, a) = \{\forall x \forall y \forall z\, (supervises(x, y) \wedge work\_in(x, z) \wedge directs(y, z)) \rightarrow \bot\}$. *This explains why the argument b attacks a.*

The information carried by the argument would be more useful if the structure exhibits the line of reasoning leading to the conclusion, called *clarifying* the argument.

Definition 3 (Clarifying $\mathbb{C}$) *Given an argument $a \in \mathscr{A}$. The mapping clarification denoted by $\mathbb{C}$ is a total function from the set $\mathscr{A}$ to $2^{\mathscr{R}}$ such that:* $\mathbb{C}(a = \langle F_0, ..., F_n \rangle) = \{r | r \in \mathscr{R}$ *s.t r is applicable to F_i and the application of r on F_i yields F_{i+1} for all $i \in \{0, ..., n\}\}$.*

Definition 4 (Clarified Argument) *Given an argument $a \in \mathscr{A}$. The corresponding clarified argument C_a is a 3-tuple $\langle \texttt{Supp}(a), \mathbb{C}(a), \texttt{Conc}(a) \rangle$ such that $\mathbb{C}(a) \subseteq \mathscr{R}$ are the rules used to derive the conclusion $\texttt{Conc}(a)$.*

Example 6 (Clarification count. Example 3) *A clarified version of the argument a is* $C_a = (\{directs(John, d_1)\}, \{\forall x \forall d\, directs(x,d) \rightarrow emp(x)\}, emp(John))\}$ *such that* $\texttt{Supp}(C_a) = \{directs(John, d_1)\}$, $\mathbb{C}(C_a) = \{\forall x \forall d\, directs(x,d) \rightarrow emp(x)\}$ *and* $\texttt{Conc}(C_a) = emp(John)$.

4. Dialectical Explanation for Query Failure

In what follows, we describe a simple dialectical system of explanation based on the work of [9]. Our system is custom-tailored for the problem of *Query Failure Explanation* under ICR-semantics in inconsistent knowledge bases with rule-based language. Our *dialectical explanation* involves two parties: the *User* and the *Reasoner*. The *User* wants to understand why the query is not ICR-entailed and the *Reasoner* provides a respond aiming at showing the reason why the query is not ICR entailed. We model this explanation through a dialogue composed of moves (speech acts) put forward by both the *User* and the *Reasoner*. This dialogue is governed by rules (pre/post conditions rules, termination rules, success rules) that specify what type of moves should follow the other, the conditions under which the dialogue terminates, and when and under which conditions the explanation has been successfully achieved (success rules).

We denote by $Arg^{+}(Q)$ the set of all arguments that support the query Q, namely $a \in Arg^{+}(Q)$ iff $\texttt{Conc}(a) \models Q$. In what follows we define types of moves that can be used in the dialogue.

Definition 5 (Moves) *A move is a 3-tuple* $m = \langle ID, I, \omega \rangle$ *such that:*

1. *m is an* ***explanation request****, denoted by* $m^{\texttt{ERQ}}$ *iff* $ID = User$, *I is a query Q and* ω *is an argument supporting Q.*
2. *m is an* ***explanation response****, denoted by* $m^{\texttt{ERP}}$ *iff* $ID = Reasoner$, *I is an argument supporting Q and* ω *is an argument such that* ω *attacks I.*
3. *m is a* ***follow-up question****, denoted by* $m^{\texttt{FQ}}$ *iff* $ID = User$, *I is an argument and* ω *is either* $\texttt{Conc}(I)$ *or an argument* ω_1 *that supports Q s.t* $(\omega, \omega_1) \in \texttt{Att}$.
4. *m is a* ***follow-up answer****, denoted by* $m^{\texttt{FA}}$ *iff* $ID = Reasoner$, *I is an argument and* ω *is either a deepening* $\mathbb{D}$ *or a clarified argument* $\mathbb{C}(I)$.

The explanation request $m^{\texttt{ERQ}} = \langle User, Q, \omega \rangle$ is an explanation request made by the *User* asking *"why the query Q is not ICR-entailed while there is an argument* ω *asserts the entailment of Q*", an explanation response $m^{\texttt{ERP}} = \langle Reasoner, \omega, \omega_1 \rangle$ made by the *Reasoner* is an explanation for the previous inquiry by showing that the argument ω (that supports Q) is the subject of an attack made by ω_1. The *User* also can ask a follow-up question if the *Reasoner* provides an explanation. The follow-up question $m^{\texttt{FQ}} = \langle User, \omega_1, \omega \rangle$ is a compound move, it can represent a need for *deepening* (the *User* wants to know why the argument ω_1 is attacking the argument ω) or the need for clarification (how the argument ω_1 comes to a certain conclusion). To distinguish them, the former has $\omega = \texttt{Conc}(\omega_1)$ and the latter has ω as an argument. A follow-up answer $m^{\texttt{FA}} = \langle Reasoner, \omega_1, \omega_1' \rangle$ is also a compound move. Actually, it depends on the follow-up question. It shows the argument ω_1 that needs to be deepened (resp. clarified) and its deepening (resp. clarification) by the deepening mapping $\mathbb{D}(\omega_1, \omega)$ (resp. clarification mapping $\mathbb{C}(\omega)$) in Definition 4 (resp. Definition 6). An example is provided afterward.

In what follows we specify the structure of dialectical explanation and the rules that have to be respected throughout the dialogue.

Definition 6 (Dialectical Explanation) *Given a QFEP $\mathscr{P}$. A dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$ is a non-empty sequence of moves $\langle m_1^s, m_2^s, ..., m_n^s \rangle$ where $s \in \{\texttt{ERQ}, \texttt{FQ}, \texttt{ERP}, \texttt{FA}\}$ and $i \in \{1, ..., n\}$ such that:*

1. *The first move is always an explanation request $m_1^{\texttt{ERQ}}$, we call it an opening.*
2. *$s \in \{\texttt{ERQ}, \texttt{FQ}\}$ iff i is odd, $s \in \{\texttt{ERP}, \texttt{FA}\}$ iff i is even.*
3. *For every explanation request $m_i^{\texttt{ERQ}} = \langle User, I_i, \omega_i \rangle$, I_i is the query Q to be explained and ω_i is an argument supporting Q and for all $m_j^{\texttt{ERQ}}$ s.t $j < i$ $\omega_i \neq \omega_j$.*
4. *For every explanation response $m_i^{\texttt{ERP}} = \langle Reasoner, I_i, \omega_i \rangle$ s.t $i \geq 1$, $I_i = \omega_{i-1}$ and $\omega_i = \omega'$ s.t $(\omega', I_i) \in \texttt{Att}$.*
5. *For every follow-up question $m_i^{\texttt{FQ}} = \langle User, I_i, \omega_i \rangle$, $i \geqslant 1$, $I_i = \omega_{i-1}$ and ω is either I_{i-1} or $\texttt{Conc}(\omega_{i-1})$.*
6. *For every follow-up answer $m_i^{\texttt{FA}} = \langle Reasoner, I_i, \omega_i \rangle$, $i \geqslant 1$, $I_i = I_{i-1}$ and $\omega_i = \mathbb{D}(I_i, \omega_{i-1})$ or $\omega = \mathbb{C}(I_i)$.*

We denote by $Arg_{user}(\mathscr{D}_{exp})$ the set of all arguments put by the User in the dialogue.

Every dialogue has to respect certain rules (protocol). Theses rules organize the way the *Reasoner* and the *User* should put the moves. For each move we specify the conditions that have to be me for the move to be valid (preconditions). We also specify the conditions that identify the next moves (postconditions).

Definition 7 (Pre/Post Condition Rules) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, $\mathscr{D}_{exp}$ has to respect the following rules:*

Explanation request:

- *Preconditions: The beginning of the dialogue or the last move of the Reasoner was either an explanation response or a follow-up answer.*
- *Postconditions: The next move must be an explanation answer.*

Explanation response:

- *Preconditions: The last move by the User was an explanation request.*
- *Postconditions: The next move must be either another explanation request (it may implicitly means that the User had not understood the previous explanation) or a follow-up question.*

Follow-up question:

- *Preconditions: The last move by the Reasoner was an explanation response or this follow-up question is not the second in a row.*
- *Postconditions: The next move must be a follow-up answer.*

Follow-up answer:

- *Preconditions: The last move by the User was a follow-up question.*
- *Postconditions: The next move must be an explanation request (it may implicitly means that the User had not understood the previous explanation).*

Beside the previous rules, there are termination rules that indicate the end of a dialectical explanation.

Definition 8 (Termination Rules) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, $\mathscr{D}_{exp}$ terminates when the User puts an empty explanation request $m_i^{\texttt{ERQ}} = \langle User, \emptyset, \emptyset \rangle$ or when $Arg_{User}(\mathscr{D}_{exp}) = Arg^{+}(Q)$.*

The rules in Definition 7 & 8 state that the *Reasoner* is always **committed** to respond with an explanation response, the *User* then may indicate the end of the dialogue by an *empty explanation request* (Definition 8) declaring his/her understanding, otherwise starts another explanation request (this indicates that he/she has not understood the last explanation) or asks a follow-up question, the *User* cannot ask more than two successive follow-up questions. If the *User* asks a follow-up question then the *Reasoner* is **committed** to a follow-up answer. When the *User* asks for another explanation he/she cannot use an argument that has already been used. If the *User* ran out of arguments and he/she has not yet understood, the dialogue ends (Definition 8) and the explanation is judged unsuccessful. It is important to notice that when the *Reasoner* wants to answer the *User* there may be more than one argument to chose. There are many "selection strategies" that can be used in such case (for instance, the shortest argument, the least attacked argument...etc), but their study is beyond the scope of the paper.

In what follows we elaborate more on the success and the failure of an explanation.

Definition 9 (Success Rules) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, $\mathscr{D}_{exp}$ is successful when it terminates with an empty explanation request $m_i^{\texttt{ERQ}} = \langle User, \emptyset, \emptyset \rangle$, otherwise it is unsuccessful.*

A dialectical explanation is judged to be successful if the *User* terminates the dialogue voluntarily by putting an empty explanation request. If the *User* has used all arguments supporting Q then he/she is forced to stop without indicating his/her understanding, in this case we consider the explanation unsuccessful. By virtue of the equivalence between ICR-semantics and argumentation presented in Section 3, the existence of response is always guaranteed. This property is depicted in the following proposition.

Proposition 1 (Existence of response) *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. Then, For every $m_i^s \in \mathscr{D}_{exp}$ s.t $s \in \{\texttt{ERQ}, \texttt{FQ}\}$ and $1 \leq i \leq |\mathscr{D}_{exp}|$, the next move m_{i+1}^s s.t $s \in \{\texttt{ERP}, \texttt{FA}\}$ always exists.*

5. Conclusion

In this paper, we have presented a dialectical approach for explaining boolean conjunctive queries failure, designated by Query Failure Explanation Problem (QFEP), in an inconsistent ontological knowledge base where inconsistency is handled by inconsistency-tolerant semantics (ICR) and issued from the set of facts. The introduced approach relies on both (i) the relation between ontological knowledge base and logical argumentation framework and (ii) the notions of argument deepening and clarifications. So, through a dialogue, the proposed approach explains to the *User* how and why his/her query is not entailed under ICR semantics.

We currently investigate the explanation problem not only for Query Failure but also for Query Answering. We have proposed a Query Explanation framework under the CoGui editor[1] and plan to test the two approaches within the DUR-DUR ANR project which investigates the use of argumentation in agri-food chains.

Acknowledgement Financial support from the French National Research Agency (ANR) for the project DUR-DUR and ASPIQ (ANR-12-BS02-0003) is gratefully acknowledged.

References

[1] A. Arioua, N. Tamani, and M. Croitoru. On conceptual graphs and explanation of query answering under inconsistency. In N. Hernandez, R. Jaesche, and M. Croitoru, editors, *Proceedings of the 2014 International Conference of Conceptual Structures (ICCS)*, volume LNAI 8577, pages 51–64. Springer, 2014.

[2] M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.

[3] M. Bienvenu and R. Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proceedings of IJCAI'13*, pages 775–781. AAAI Press, 2013.

[4] A. Borgida, D. Calvanese, and M. Rodriguez-Muro. Explanation in the dl-lite family of description logics. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *LNCS*, pages 1440–1457. Springer Berlin Heidelberg, 2008.

[5] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:57–83, 2012.

[6] D. Calvanese, M. Ortiz, M. Šimkus, and G. Stefanoni. Reasoning about explanations for negative query answers in dl-lite. *Journal of Artificial Intelligence Research*, 48:635–669, 2013.

[7] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5):286–310, 2007.

[8] M. Croitoru and S. Vesic. What can argumentation do for inconsistent ontology query answering? In W. Liu, V. Subrahmanian, and J. Wijsen, editors, *Scalable Uncertainty Management*, volume 8078 of *LNCS*, pages 15–29. Springer Berlin Heidelberg, 2013.

[9] Douglas Walton. A Dialogue System Specification for Explanation . *Synthese, 182(3), 2011,*, Volume 182(3):349–374, 2011.

[10] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[11] A. J. Garcia, C. I. Chesñevar, N. D. Rotstein, and G. R. Simari. Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications*, 40(8):3233 – 3247, 2013.

[12] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems*, RR'10, pages 103–117, Berlin, Heidelberg, 2010. Springer-Verlag.

[13] T. Lukasiewicz, M. V. Martinez, and G. I. Simari. Complexity of inconsistency-tolerant query answering in datalog+/-. In R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, P. D. Leenheer, and D. Dou, editors, *Proceedings of the 12th International Conference on Ontologies, Databases, and Applications of Semantics, September 10-11, 2013*, volume 8185 of *LNCS*, pages 488–500. Springer, 2013.

[14] D. L. McGuinness and P. F. Patel-Schneider. Usability issues in knowledge representation systems. In *In Proc. of AAAI-98*, pages 608–614, 1998.

[15] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. In *Journal on data semantics X*, pages 133–173. Springer, 2008.