Query Failure Explanation in Inconsistent Knowledge Bases: A Dialogical Approach

Abdallah Arioua, Nouredine Tamani, Madalina Croitoru, Patrice Buche

Query Failure Explanation in Inconsistent Knowledge Bases: A Dialogical Approach

Abdallah Arioua, Nouredine Tamani, Madalina Croitoru and Patrice Buche

Abstract In the EcoBioCap project (www.ecobiocap.eu) about the next generation of packaging, a decision support system has been built that uses argumentation to deal with stakeholder preferences. However, when testing the tool the domain experts did not always understand the output of the system. The approach developed in this paper is the first step to the construction of a decision support system endowed with an explanation module. We place ourselves in the equivalent setting of inconsistent Ontology-Based Data Access (OBDA) and addresses the problem of explaining Boolean Conjunctive Query (BCQ) failure. 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 inconsistency tolerant 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 [19]. In

Abdallah Arioua
IATE, INRA,UM2, 2 place Pierre Viala, 34060 Montpellier,
e-mail: abdallaharioua@gmail.com

Nouredine Tamani
INRIA GraphIK, LIRMM, 161 rue Ada, 34095 Montpellier,
e-mail: tamani@lirmm.fr, ntamani@gmail.com

Madalina Croitoru
INRIA GraphIK, LIRMM, UM2, 161 rue Ada, 34095 Montpellier, e-mail: croitoru@lirmm.fr

Patrice Buche
IATE, CIRAD-INRA-Supagro-UM2, 2 place Pierre Viala, 34060 Montpellier,
e-mail: patrice.buche@supagro.inra.fr

practical systems involving large amounts of data and multiple data sources, data inconsistency with respect to the ontology is difficult to prevent. Many inconsistency-tolerant semantics [5, 4, 16, 17] have been proposed that rely on the notion of data repairs i.e. subsets of maximally consistent data with respect to the ontology. Different query answering strategies over these repairs (called semantics) are investigated in the literature. For instance, computing answers that hold in every repair corresponds to AR-semantics, computing answers that hold under the intersection closed repairs corresponds to ICR-semantics, etc.

In this paper we will consider inconsistent knowledge bases and focus on the ICR-semantics (**I**ntersection of **C**losed **R**epair). The particular choice of the ICR-semantics for OBDA is due to its interesting productivity properties as shown in [4]. Under the ICR-semantics, maximal (in terms of set inclusion) consistent subsets (called repairs) of the knowledge base are constructed. Querying the knowledge base is done on the intersection of repairs after ontology closure.

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 [18]. Moreover, as argued by Calvanese et al. in [10] *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), which is the research problem addressed in this paper. Precisely: "*Given an inconsistent knowledge base, denoted by KB, and a boolean conjunctive query Q, why Q is not entailed from KB under the ICR-semantics?*".

To address this issue we use argumentation as an approach for explanation. We consider the logical instantiation of Dung's [14] abstract argumentation framework for OBDA in [12] 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 (an introduced) language-based primitives such as *clarification* and *deepening* to refine the answer.

This paper improves over and extends the work presented in [1]. Its contribution lies in the following points: first, it puts forward the explanation power of argumentation in the benefit of Semantic Web. Second, it improves OBDA system's usability and enhances its user-friendliness. To the best of our knowledge, 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 [6, 10] in handling query failure since we consider an inconsistent setting within OBDA. In addition, the work presented in [15] is neither applied to an OBDA context nor to the Datalog+/- language.

The paper is organized as follows. In Section 2, we recall the scope of the paper and formally define the addressed problem using an illustrating example. In Section 3 and 4, we present the language and the corresponding argumentation framework. In Section 5, we introduce the formalization of the proposed explanation and show several properties. Finally, Section 6 concludes the paper.

2 Paper in a Nutshell

Let us introduce the motivation and the formal context of our work.
The Context. Several semantics that allow answering queries in the presence of inconsistency have been studied in the literature and here we only focus on the ICR-semantics.
The 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". Such query failure might occur, for instance, after the *User* does not find in the answer set the expected output (e.g. the *User* asked for all the students and Luca did not appear in the answer list). Another possibility is when the *User* asks for the existence of students in the knowledge base and the system answers that no students are present in the knowledge base.

Observe that the answer 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 on 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 $\mathscr{K} \nvDash_{ICR} Q$. We call $\mathscr{P} = \langle \mathscr{K}, Q \rangle$ a Query Failure Explanation Problem (QFEP).

The Guidelines of the Solution. To address the *Query Failure Explanation Problem*, we use a logical instantiation of Dung's [14] abstract argumentation framework for OBDA in [12] which ensures that the argumentation framework used throughout the paper respects the rationality postulates [11]. The arguments are then used for query failure explanation as follows. First, we introduce two notions over the arguments: clarifying and deepening. The former notion serves at unfolding the knowledge used in the argument, and the latter is used to explicit the reason for an attack between two arguments. We then describe a dialectical system of explanation custom-tailored for inconsistent knowledge base query failure. The *User* and the *Reasoner* will take turns in a dialogue with the final aim that the *User* understands why a query is not entailed under the ICR-semantics by the knowledge base.

3 Formal Settings

In this section, we introduce: (1) OBDA setting and representation language, (2) the inconsistency-tolerant semantics.

3.1 OBDA Setting

There are two major approaches to represent an ontology for the OBDA problem: Description Logics (such as $\mathscr{EL}$ [2] and DL-Lite [9] families) and rule-based languages (such as Datalog+/- [7] language, a generalization of Datalog that allows for existentially quantified variables in rules heads). Despite Datalog+/- undecidability when answering conjunctive queries, different decidable fragments of Datalog+/- are studied in the literature [3]. These fragments generalize the aforementioned Description Logics families and overcome their limitations by allowing any predicate arity as well as cyclic structures. Here we follow the second method and use a general rule-based setting knowledge representation language equivalent to the Datalog+/- language.

3.2 Language Specification

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 $(\exists,\forall)$. We consider first-order vocabularies with constants but no other function symbol. A term t is a constant or a variable, different constants represent different values (unique name assumption), an atomic formula (or atom) is of the form $p(t_1,...,t_n)$ where p is an n-ary predicate, and $t_1,...,t_n$ are terms. A *ground* atom is an atom with no variables. A variable in a formula is free if it is not in the scope of a quantifier. A formula is closed if it has no free variables. We denote by $\mathbf{X}$ (with a bold font) a sequence of variables $X_1,...,X_k$ with $k \geq 1$. A *conjunct* $C[\mathbf{X}]$ is a finite conjunction of atoms, where $\mathbf{X}$ is the sequence of variables occurring in C. Given an atom or a set of atoms A, $vars(A)$, $consts(A)$ and $terms(A)$ denote its set of variables, constants and terms, respectively.

An *existential rule* (or simply a rule) is a first-order formula of the form $r = \forall\mathbf{X}\,\forall\mathbf{Y}\,(H[\mathbf{X},\mathbf{Y}]) \rightarrow \exists\mathbf{Z}\,C[\mathbf{Z},\mathbf{Y}]$, with $vars(H) = \mathbf{X}\cup\mathbf{Y}$, and $vars(C) = \mathbf{Z}\cup\mathbf{Y}$ where H and C are *conjuncts* called the hypothesis and conclusion of r respectively. We denote by $r = (H,C)$ a contracted form of a rule r. An existential rule with an empty hypothesis is called a *fact*. A fact is an existentially closed (with no free variable) conjunct. i.e. $\exists x(teacher(x) \wedge employee(x))$.

We recall that a *homomorphism* π from set of atoms A_1 to set of atoms A_2 is a substitution of $vars(A_1)$ by $terms(A_2)$ such that $\pi(A_1) \subseteq A_2$. Given two facts f and f'. We have $f \models f'$ if and only if there is a homomorphism from f' to f, where $\models$ is the first-order semantic entailment.

A *rule* $r = (H,C)$ is *applicable* to a set of facts F if and only if there exists $F' \subseteq F$ such that there is a homomorphism π from H to the conjunction of elements of F'. For instance, the rule $\forall x(teacher(x) \rightarrow employee(x))$ is applicable to the set $\{teacher(Tom), cute(Tom)\}$ because there is a homomorphism from $teacher(x)$ to $teacher(Tom)$. If a rule r is applicable to a set of facts F, its application according to π produces a set $F \cup \{\pi(C)\}$. The new set $F \cup \{\pi(C)\}$, denoted also by $r(F)$, is called *immediate derivation* of F by r. In our example the produced set (immediate derivation) is $\{teacher(Tom), employee(Tom), cute(Tom)\}$.

A *negative constraint* (or constraint) is a first-order formula $n = \forall \mathbf{X}\, H[\mathbf{X}] \rightarrow \perp$ where $H[\mathbf{X}]$ is a conjunct called hypothesis of n and $\mathbf{X}$ is a sequence of variables appearing in the hypothesis. For example, $n = \forall x \forall y \forall z (supervises(x,y) \wedge work\_in(x,z) \wedge directs(y,z)) \rightarrow \perp$, means that it is impossible for x to supervise y if x works in department z and y directs z.

The Knowledge Base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ consists of a finite set of facts $\mathscr{F}$, a finite set of existential rules $\mathscr{R}$ and a finite set of negative constrains $\mathscr{N}$.

Example 1. The following example is inspired from [8]. 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 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:

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

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

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

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

$\mathscr{R}$-derivation. 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 such that $F_0 = F$, and for all $i \in \{0, ..., n-1\}$ there is a rule $r_i = (H_i, C_i) \in \mathscr{R}$ and a homomorphism π_i from H_i to F_i such that $F_{i+1} = F_i \cup \{\pi(C_i)\}$. For a set of facts $F \subseteq \mathscr{F}$ and a query Q and a set of rules $\mathscr{R}$, we say $F, \mathscr{R} \models Q$ if and only if there exists an $\mathscr{R}$-derivation $\langle F_0, ..., F_n \rangle$ such that $F_n \models Q$.

Closure. 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 defined as the smallest set (with respect to $\subseteq$) which contains F and is closed under $\mathscr{R}$-derivation.

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 $F, \mathscr{R} \models f$) if and only if the closure of F by all the rules entails f (i.e. $\mathtt{Cl}_{\mathscr{R}}(F) \models f$).

A conjunctive query (CQ) has the form $Q(\mathbf{X}) = \exists \mathbf{Y} \Phi[\mathbf{X},\mathbf{Y}]$ where $\Phi[\mathbf{X},\mathbf{Y}]$ is a conjunct such that $\mathbf{X}$ and $\mathbf{Y}$ are variables appearing in Φ. A boolean CQ (BCQ) is a CQ of the form $Q()$ with an answer yes or no, e.g. $Q = \exists x\, emp(x)$. We refer to a BCQ with an answer *no* as **failed query**, whereas a query with the answer *yes* as **accepted query**. Unless stated otherwise, we refer to a BC query as a query.

3.3 Inconsistency-Tolerant Semantics

Given a knowledge base $\mathscr{K} = (\mathscr{F},\mathscr{R},\mathscr{N})$, a set $F \subseteq \mathscr{F}$ is said to be *inconsistent* if and only if 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 if and only if it is not inconsistent. A set $F \subseteq \mathscr{F}$ is $\mathscr{R}$*-inconsistent* if and only if there exists a constraint $n \in \mathscr{N}$ such that $\mathtt{Cl}_{\mathscr{R}}(F) \models H_n$. A set of facts is said to be $\mathscr{R}$*-inconsistent* if and only if it is not $\mathscr{R}$*-consistent*. A knowledge base $(\mathscr{F},\mathscr{R},\mathscr{N})$ is said to be *inconsistent* if and only if $\mathscr{F}$ is $\mathscr{R}$-inconsistent.

Notice that (like in classical logic), if a knowledge base $\mathscr{K} = (\mathscr{F},\mathscr{R},\mathscr{N})$ is inconsistent, then everything is entailed from it. A common solution [4, 16] 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) [4].

Definition 2 (ICR-semantics). Let $\mathscr{K} = (\mathscr{F},\mathscr{R},\mathscr{N})$ be a knowledge base and let Q be a query. Q is ICR-entailed from $\mathscr{K}$, written $\mathscr{K} \models_{ICR} Q$ if and only if:

$$\bigcap_{\mathscr{A} \in \mathscr{R}epair(\mathscr{K})} \mathtt{Cl}_{\mathscr{R}}(\mathscr{A}) \models Q.$$

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). We can conclude that the ICR-semantics is prudent and operates according to the principle “when in doubt, throw it out”.

4 Argumentation Framework, Deepening and Clarification

In what follows we present the definition of argumentation framework in the context of rule-based languages. We use the definition of argument of [12] and extend it to the notions of deepened and clarified arguments.

4.1 Rule-Based Dung Argumentation Framework

As defined in [12], given a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, the *corresponding argumentation framework* $\mathscr{A}\mathscr{F}_{\mathscr{K}}$ is a pair $(\mathtt{Arg}, \mathtt{Att})$ where $\mathtt{Arg}$ is the set of arguments that can be constructed from $\mathscr{F}$ and $\mathtt{Att}$ is an asymmetric binary relation called *attack* defined over $\mathtt{Arg} \times \mathtt{Arg}$. An argument is defined as follows.

Definition 3 (Argument [12]). Given a knowledge base $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$, an argument a is a tuple $a = (F_0, F_1, \ldots, F_n, C)$ where:
• $(F_0, \ldots, F_n)$ is an $\mathscr{R}$-derivation of F_0 in $\mathscr{K}$,
• C is an atom, a conjunction of atoms, the existential closure of an atom or the existential closure of a conjunction of atoms such that $F_n \models C$.
F_0 is the support of the argument a (denoted $\mathtt{Supp}(a)$) and C is its conclusion (denoted $\mathtt{Conc}(a)$).

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

Definition 4 (Attack [12]). The attack between two arguments expresses the conflict between their conclusion and support. An argument a attacks an argument b iff there exists a fact $f \in \mathtt{Supp}(b)$ such that $\{\mathtt{Conc}(a), f\}$ is $\mathscr{R}$-inconsistent.

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.

4.1.1 Admissibility, Semantics and Extensions

Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a knowledge base and $\mathscr{A}\mathscr{F}_{\mathscr{K}}$ its *corresponding argumentation framework*. Let $\mathscr{E} \subseteq \mathtt{Arg}$ be a set of arguments. We say that $\mathscr{E}$ is *conflict free* if and only if there exist no arguments $a, b \in \mathscr{E}$ such that $(a, b) \in \mathtt{Att}$. $\mathscr{E}$ *defends* an argument a if and only if for every argument $b \in \mathtt{Arg}$, if we have $(b, a) \in \mathtt{Att}$ then there exists $c \in \mathscr{E}$ such that $(c, b) \in \mathtt{Att}$. $\mathscr{E}$ is *admissible* if and only if it is conflict free and defends all its arguments. $\mathscr{E}$ is a *preferred extension* if and only if it is maximal (with respect to set inclusion) admissible set (please see [14] for other types of semantics).

We denote by $\texttt{Ext}(\mathscr{AF}_{\mathscr{K}})$ the set of all extensions of $\mathscr{AF}_{\mathscr{K}}$ under one Dung's semantics. An argument is sceptically accepted if it is in all extensions, credulously accepted if it is in at least one extension and not accepted if it is not in any extension.

4.1.2 Equivalence between ICR and Preferred semantics

Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a knowledge base and $\mathscr{AF}_{\mathscr{K}}$ the corresponding argumentation framework. A query Q is sceptically accepted under preferred semantics if and only if $(\bigcap_{\mathscr{E} \in \texttt{Ext}(\mathscr{AF}_{\mathscr{K}})} \texttt{Concs}(\mathscr{E})) \vDash Q$, such that $\texttt{Concs}(\mathscr{E}) = \bigcup_{a \in \mathscr{E}} \texttt{Conc}(a)$. The results of [12] show the equivalence between sceptically acceptance under preferred semantics and ICR-entailment:

Theorem 1 (Semantics equivalence [12]). *Let $\mathscr{K} = (\mathscr{F}, \mathscr{R}, \mathscr{N})$ be a knowledge base, let $\mathscr{AF}_{\mathscr{K}}$ the corresponding argumentation framework, Q a query. $\mathscr{K} \models_{ICR} Q$ if and only if Q sceptically accepted under preferred semantics.*

4.2 Deepening and Clarifying

In the following we show some functionalities that give the *User* the possibility to manipulate arguments to gain clarity, 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 5 (Deepening $\mathbb{D}$). Given two arguments $a, b \in \texttt{Arg}$. The mapping *deepening* denoted by $\mathbb{D}$ is a total function from the set $\texttt{Att}$ to $2^{\mathscr{N}}$ defined as follows:
$\mathbb{D}(b,a) = \{nc|$ 1. $nc \in \mathscr{N}$ and ,
2. $\exists f \in \texttt{Supp}(a)$ such that $\texttt{Cl}_{\mathscr{R}}(\{Conc(b), f\}) \models H_{nc}$.
$\}$, where H_{nc} is the hypothesis of the constraint nc.

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. Deepening is $\mathbb{D}(b,a) = \{\forall x \forall y \forall z\,(supervises(x,y) \wedge work\_in(x,z) \wedge directs(y,z)) \rightarrow \bot\}$ which 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 that leads to the conclusion. We call this *clarifying* the argument.

Definition 6 (Clarifying $\mathbb{C}$). Given an argument $a \in \texttt{Arg}$. The mapping *clarification* denoted by $\mathbb{C}$ is a total function from the set $\texttt{Arg}$ to $2^{\mathscr{R}}$ such that: $\mathbb{C}(a = \langle F_0, ..., F_n, C\rangle) = \{r | r \in \mathscr{R}$ such that r is applicable to F_i and the application of r on F_i yields F_{i+1} for all $i \in \{0, ..., n-1\}\}$.

Definition 7 (Clarified Argument). Given an argument $a \in \texttt{Arg}$. 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 $C_a = (\{diretcs(John, d_1)\}, \{\forall x \forall d\, directs(x,d) \rightarrow emp(x)\}, emp(John))\}$ such that $\texttt{Supp}(a) = \{directs(John, d_1)\}$, $\mathbb{C}(a) = \{\forall x \forall d\, directs(x,d) \rightarrow emp(x)\}$ and $\texttt{Conc}(a) = emp(John)$.

Relying on these notions, in the next section, we provide an argumentation-based explanation called *dialectical-explanation* based on Walton's *Dialogical Model of Explanation* [21, 20, 13]. In this type of dialogue an exchange of information takes place between an Explainer and Explainee with the purpose of explaining certain assertions.

5 Dialectical Explanation for Query Failure

In what follows we describe a simple dialectical system of explanation based on the work of [21, 20, 13]. 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* (the explainee) and the *Reasoner* (the explainer). The *User* wants to understand why the query is not ICR-entailed and the *Reasoner* provides a response 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*.

Example 7 (Motivating Example Count. Example 1). Consider the following conjunctive query $Q(x) = \forall x\, works\_in(x, Statistics)$ stands for "Get me all the employees in the department of statistics". The system returns as a result the set of answers $A = \{Carlo, Luca, Jane\}$ under ICR-semantics. The *User* expected that the query $Q(Linda) = Works\_in(Linda, Statistics)$ is also ICR-entailed, that means "Linda" is a part of the answers of $Q(x)$. In addition, the *User* knows also a set of arguments that support his/her expectation denoted by $Arg^{+}(Q(Linda)) = \{\omega\}$ such that $\omega = (\{works\_in(Linda, Statistics)\}, works\_in(Linda, Statistics))$ which stands for "Linda is an employee in the department of statistics since it is specified as a fact". Now the *User* wants to understand why $Q(Linda)$ is not ICR-entailed. A dialectical explanation is represented as follows:

1. $User$: Why not $Q(Linda)$ given that ω ?
2. $Reasoner$: Because $(\{Works\_in(Linda, Accounting)\}, Works\_in(Linda, Accounting))$ (is a counterargument denoted by ω_1)
3. $User$: Why $Works\_in(Linda, Accounting)$?
4. $Reasoner$: Because $uses\_\text{office}(Linda, o_1)$ and $locate\_in(o_1, Accounting)$ so $works\_in(Linda, Accounting)$
5. $User$: how's that can be a problem ?

6. *Reasoner*: The following negative constraint is violated
$n_3 = \forall x \forall y \forall z\ (works\_in(x,y) \wedge works\_in(x,z)) \rightarrow \bot$.

We denote by $Arg^+(Q)$ the set of all arguments that support the query Q, namely $a \in Arg^+(Q)$ if and only if $\texttt{Conc}(a) \models Q$.

This dialogue is governed by rules (pre/post conditions, termination and 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).

In what follows we define types of moves that can be used in the dialogue.

Definition 8 (Moves). A move is a 3-tuple $m = \langle ID, I, \omega \rangle$ such that:

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

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

Definition 9 (Dialectical Explanation). Given a QFEP $\mathscr{P}$. A dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$ is a non-empty sequence of moves $\langle m_1^s, m_2^s, \ldots, m_n^s \rangle$ where $s \in \{\texttt{ERQ}, \texttt{FQ}, \texttt{ERP}, \texttt{FA}\}$ and $i \in \{1, \ldots, n\}$ such that:

1. The first move is always an explanation request $m_1^{\texttt{ERQ}}$, we call it an opening.
2. $s \in \{\texttt{ERQ}, \texttt{FQ}\}$ if and only if i is odd, $s \in \{\texttt{ERP}, \texttt{FA}\}$ if and only if i is even.

3. For every explanation request $m_i^{\mathtt{ERQ}} = \langle User, I_i, \omega_i \rangle$, I_i is the query Q to be explained and ω_i is an argument supporting Q and for all $m_j^{\mathtt{ERQ}}$ such that $j < i$ $\omega_i \neq \omega_j$.
4. For every explanation response $m_i^{\mathtt{ERP}} = \langle Reasoner, I_i, \omega_i \rangle$ such that $i \geq 1$, $I_i = \omega_{i-1}$ and $\omega_i = \omega'$ such that $(\omega', I_i) \in \mathtt{Att}$.
5. For every follow-up question $m_i^{\mathtt{FQ}} = \langle User, I_i, \omega_i \rangle$, $i \geqslant 1$, $I_i = \omega_{i-1}$ and ω is either I_{i-1} or $\mathtt{Conc}(\omega_{i-1})$.
6. For every follow-up answer $m_i^{\mathtt{FA}} = \langle Reasoner, I_i, \omega_i \rangle$, $i \geqslant 1$, $I_i = I_{i-1}$ 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.

As stated, a dialectical explanation is a sequence of moves put forward by either the *User* or the *Reasoner*. In the following example we model the dialectical explanation of Example 7 and we show the encoding of different moves.

Example 8 (Cont. Example 7). The opening in line (1) (first move) is an explanation request $m_1^{\mathtt{ERQ}} = \langle User, Q(Linda), \omega \rangle$, the *Reasoner* responded with an explanation response $m_2^{\mathtt{ERP}} = \langle Reasoner, \omega, \omega_1 \rangle$ then the *User* asked a follow-up question seeking clarification $m_3^{\mathtt{FQ}} = \langle User, \omega_1, \mathtt{Conc}(\omega_1) \rangle$ to know why $\mathtt{Conc}(\omega_1)$="Linda is working in department of Accounting". The *Reasoner* responded with a follow-up answer for clarification $m_4^{\mathtt{FA}} = \langle Reasoner, \omega_1, \mathbb{C}(\omega_1) \rangle$ showing a line of reasoning that led to the conclusion $\mathtt{Conc}(\omega_1)$. Another type of follow-up question is presented in line (5) in which the *User* asked a follow-up question for deepening $m_5^{\mathtt{FA}} = \langle Reasoner, \omega_1, \omega \rangle$ wondering how the fact that "working in department of Accounting contradicts that Linda works in Department of statistics". The *Reasoner* responded with a follow-up answer $m_6^{\mathtt{FA}} = \langle Reasoner, \omega_1, \mathbb{D}(\omega_1, \omega) \rangle$ such that $\mathbb{D}(\omega_1, \omega)$)="An employee cannot work in two different departments at once". The key point of such responses is that it reflects the intuition described in Section 2 about ICR-semantics which dictates that if there is a conflicting information about the query Q then it cannot be deduced.

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

Definition 10 (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 11 (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 $m^{\mathtt{ERQ}} = \langle User, \emptyset, \emptyset \rangle$ or when $Arg_{User}(\mathscr{D}_{exp}) = Arg^{+}(Q)$.

The rules in Definition 9, 10 and 11 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 11) 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 11) 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 choose. 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 rules.

Definition 12 (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 $m_i^{\mathtt{ERQ}} = \langle User, \emptyset, \emptyset \rangle$, 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 $m_i^s \in \mathscr{D}_{exp}$ such that $s \in \{\mathtt{ERQ}, \mathtt{FQ}\}$ and $1 \leq i \leq |\mathscr{D}_{exp}|$, the next move m_{i+1}^s such that $s \in \{\mathtt{ERP}, \mathtt{FA}\}$ always exists.*

Proof. For the move $m_i^{\mathtt{ERQ}} = \langle User, Q, \omega \rangle$ the query Q is not ICR-entailed therefore the argument ω that supports Q is not sceptically accepted, hence there is always an argument ω' such that $(\omega', \omega) \in \mathtt{Att}$. Thus we can construct the following explanation response move: $m_{i+1}^{\mathtt{ERP}} = \langle Reasoner, \omega, \omega' \rangle$ such that $s \in \{\mathtt{ERP}\}$.

For the move $m_i^{\mathtt{FA}}$ the proof is immediate since the mappings *deepening* $\mathbb{D}$ and *clarification* $\mathbb{C}$ are total functions.

Another issue is the finiteness of the dialectical explanation. It is not hard to conclude that a dialectical explanation is finite (from success and termination rules).

Proposition 2 (Finiteness). *Given a QFEP $\mathscr{P}$ and a dialectical explanation $\mathscr{D}_{exp}$ for $\mathscr{P}$. $\mathscr{D}_{exp}$ is finite.*

6 Conclusion

In this paper we have presented a dialogical 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.

For future work, we aim at generalizing the explanation to general conjunctive queries and studying the proposed explanation in the context of other inconsistency-tolerant semantics.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/ 2007-2013) under the grant agreement n°FP7-265669-EcoBioCAP project.

References

1. A. Arioua, N. Tamani, and M. Croitoru. Query failure explanation in inconsistent knowledge bases an argumentation approach: Extended abstract. In *5th International Conference on Computational Models of Argument 2014, to appear*, 2014.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the el envelope. In *Proc. of IJCAI 2005*, 2005.
3. J.-F. Baget, M.-L. Mugnier, S. Rudolph, and M. Thomazo. Walking the complexity lines for generalized guarded existential rules. In *Proceedings of IJCAI'11*, pages 712–717, 2011.

4. M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc of AAAI*, 2012.
5. 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.
6. 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.
7. A. Calì, G. Gottlob, and T. Lukasiewicz. Datalog$+/-$: a unified approach to ontologies and integrity constraints. In *Proceedings of the 12th International Conference on Database Theory*, pages 14–30. ACM, 2009.
8. 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.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
10. 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.
11. M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5):286–310, 2007.
12. 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.
13. Douglas Walton. A Dialogue System Specification for Explanation . *Synthese, 182(3), 2011,*, Volume 182(3):349–374, 2011.
14. 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.
15. 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.
16. 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.
17. 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.
18. D. L. McGuinness and P. F. Patel-Schneider. Usability issues in knowledge representation systems. In *In Proc. of AAAI-98*, pages 608–614, 1998.
19. 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.
20. T. Roth-Berghofer, S. Schulz, D. Bahls, and D. B. Leake, editors. *Explanation-Aware Computing, Papers from the 2007 AAAI Workshop, Vancouver, British Columbia, Canada, July 22-23, 2007*, volume WS-07-06 of *AAAI Technical Report*. AAAI Press, 2007.
21. D. Walton. A new dialectical theory of explanation. *Philosophical Explorations*, 7(1):71–89, Mar. 2004.