

Explanatory Dialogues with Argumentative Faculties over Inconsistent Knowledge Bases

Abdallah Arioua, Patrice Buche

*UMR IATE. French National Institute for Agricultural Research.
Place Pierre Viala, 34060 Montpellier, France.
abdallaharioua@gmail.com, buche@inra.fr.*

Madalina Croitoru

*University of Montpellier.
LIRMM, 161 rue ADA, F34392 Montpellier Cedex 5, Montpellier, France
croitoru@lirmm.fr.*

Abstract

We introduce a formal model of explanatory dialogue called EDS. We extend this model by including argumentation capacities to facilitate knowledge acquisition in inconsistent knowledge bases. To prove the relevance of such model we provide the DALEK (DiALectical Explanation in Knowledge-bases) framework that implements this model. We show the usefulness of the framework on a real-world application in the domain of Durum Wheat sustainability improvement within the ANR (French National Agency) funded Dur-Dur project. The preliminary pilot evaluation of the framework with agronomy experts gives a promising indication on the impact of explanation dialogues on the improvement of the knowledge's content.

Keywords: Knowledge Representation and Reasoning, Explanation, Argumentation, Dialogue, Formal Dialectics, Logic-based Argumentation

*Corresponding author

Email address: abdallaharioua@gmail.com (Abdallah Arioua)

1. Introduction

In the popular Ontology-based Data Access setting the domain knowledge is represented by an ontology facilitating query answering over existing data (Poggi et al., 2008). In practical OBDA systems involving large amounts of data and multiple data sources, data inconsistency might occur (Lembo et al., 2015). In the literature, such inconsistency is addressed by reparation techniques: the extraction of maximal consistent subsets and reasoning over them (Lembo et al., 2015; Bourgaux, 2016; Du & Qi, 2015; Bienvenu & Rosati, 2013; Bienvenu, 2012; Lembo et al., 2010) (i.e. considering their intersection, the intersection of the closure, etc.). While such strategies ensure quality query answering (at a high computational cost (Lukasiewicz et al., 2015)) they only keep the consistent and contradiction-free subsets of knowledge. This approach is too drastic as it **removes** a lot of expert knowledge. It would be more fertile to acquire more knowledge from experts in a *rule-governed and structured* way to potentially solve some sources of inconsistency. This paper paves the way for such solution.

The motivation of our work stems also from a practical aspect. In the Dur-Dur research project¹ we aim at restructuring the Durum Wheat agrifood chain in France by reducing pesticide and fertilizer usage while providing a protein-rich Durum Wheat. The project relies on constructing a *Datalog±* (Cali et al., 2012) multidisciplinary knowledge base (involving all actors in the agrofood chain) which will be used as a reference for decision making. This knowledge base is collectively built by several knowledge engineers from different sites of the project. Due to various causes (errors in the factual information due to typos, erroneous databases / Excel files, incomplete facts, unspoken obvious information “everybody knows” etc.) the collectively built knowledge base (KB) is prone to *inconsistencies*. Applying classical repairing strategies will result in a loss of considerable amount of acquired knowledge. Consequently, this would result in an inefficient exploitation of time and resources which were allocated to

¹<http://www.agence-nationale-recherche.fr/?Projet=ANR-13-ALID-0002>.

the knowledge engineer in the project. Therefore, more conservative repairing strategies are needed.

The main salient point of the paper is proposing a formal model of explanatory dialogue used for the acquisition of new knowledge to remove inconsistencies. We build on Preece (1993) and focus on improving a *prototypical* knowledge base. We propose the EDS formal model of explanatory dialogue that takes place between the domain expert and the system that explains *query entailment* in inconsistent knowledge bases in order to better expose its content to the expert. We demonstrate how the model’s instantiation is concretely used in agronomy for improving an inconsistent knowledge base. Our hypothesis is that using a formal model of explanation increases the acquired expert knowledge and removes inconsistencies.

The contribution of the paper is summarized hereafter:

- We propose a formal model of explanatory dialogues that integrates questions and argumentation (as opposed to Walton (2016); Arioua & Croitoru (2015)).
- We show how the model is used in the acquisition of new knowledge in inconsistent knowledge bases to remove inconsistencies.
- We provide a general schema to implement such model.

We implemented a dialogue framework called DALEK (**Di**ALectical **E**xplanation in **K**nowledge-bases) and carried out a pilot evaluation with agronomy experts. The goal of this evaluation is to measure the quantity of acquired knowledge and reduced inconsistencies in the presence/absence of explanation dialogues. Our hypothesis is that using explanatory dialogues increases the acquired expert knowledge and removes inconsistencies.

To show the significance and the motivation of our work consider the following example which is an excerpt of a real dialogue within the project Dur-Dur².

²This dialogue is carried out within the pilot evaluation which will be detailed in Section

Example 1 (Motivating Example). Consider the query =“ Do we perform stubble breaking?” which was asked by the **User** and to which the **Reasoner** has answered yes. The following is an explanatory dialogue explaining why we perform stubble breaking.

1. **User:** *Why do we perform stubble breaking?*
2. **Reasoner:** *Stubble breaking is necessary to prepare the soil.*
3. **User:** *Why do we prepare the soil?*
4. **Reasoner:** *Because there was a cultural precedent on the soil.*
5. **User:** *Which cultural precedent?*
6. **User:** *Sunflower.*
7. **User:** *I still don't understand why do we need to prepare the soil.*
8. **Reasoner:** *The soil needs to be soften.*
9. **User:** *Okey , I understand.*
10. **User:** *But I still don't understand why do we perform stubble breaking.*
11. **Reasoner:** *Can you tell me what are the hazards against Durum Wheat?*
12. **User:** *Fungi, viruses, weed.*
13. **Reasoner:** *Exactly, stubble breaking is used against weed. By softening the soil, the seeds of the weed will germinate and grow. Consequently, they will be easy to eliminate in the next seeding plan.*
14. **User:** *Ok, I understand now why we perform stubble breaking.*

6. Note that the **Reasoner** uses the Durum Wheat knowledge base that has been constructed for the project.

The dialogue starts at stage (1) by an explanation request or a why-question about a factual statement addressed to the **Reasoner** by the **User**. At stage (2) the **Reasoner** attempts to explain why do we perform stubble breaking. This attempt has evoked another explanation request at stage (3) to which the **Reasoner** attempted to explain at stage (4). This attempt, in its turn, has evoked a follow-up question (which-question) which is asked by the **User** at stage (5). The **User** asks such question to better understand the explanation made at stage (4). Unfortunately, after having received an answer to his/her question at stage (6), the **User** at stage (7) is still unable to understand why one needs to prepare the soil. The **Reasoner** keeps trying to explain at stage (8) where it provides another attempt to explain why one needs to prepare the soil. Note that at this stage the scope is not on the first why-question, but rather on another evoked explanation request which depends on the first one. Fortunately, at stage (9) the **User** comes to understand why we need to prepare the soil. However, this does not necessarily mean that the **User** has understood why do we perform stubble breaking. That is why at stage (10) he/she disacknowledges understanding. At this moment the **Reasoner** has chosen another way to explain. It makes use of question at stage (11) where it asks the **User** for Durum Wheat hazards, then it uses the answer of the **User** to give another attempt at stage (13). Now the **User** finds this explanation intelligible. Consequently he/she acknowledges understanding and the dialogue ends.

As one should notice, this dialogue aims at making the **User** understand the entailment of a query from the **Reasoner**'s knowledge base. Thus this dialogue has a pure explanatory purpose. However, since the **Reasoner**'s knowledge base may have some inconsistencies the feedback from the **User** is very important.

Imagine another course of action of the stage (14) where the **User** opposes to the explanation at stage (13) as follows:

15. **User**: This will postpone seeding the Durum Wheat, which is against of what we want to do on this soil (early seeding).

16. **Reasoner**: I concede.

The dialogue becomes argumentative where the **User** opposes to the **Reasoner** by giving an argument that attacks its explanation. It is clear from the **Reasoner**'s position that we will do a late seeding (a consequence of stubble breaking). This position is inconsistent with the **User**'s position. Therefore, it is assumed to be wrong since the **User** is an expert. This means that the **Reasoner** should not infer such conclusion. Since the **Reasoner** could not counterattack the **User**, although it is allowed to do so, it concedes to the **User**'s position. From this point, the dialogue can continue, either by the **Reasoner** proving another explanation, or the **Reasoner** declaring inability to explain; or by the **User** in acknowledging or disacknowledging understanding.

This type of explanatory dialogue is corrective, it in fact allows us to consider the feedback of the **User** in reducing the inference of inconsistent conclusions by exposing the content of the knowledge base in a rule-governed and goal-directed manner. It has permitted to show only the information relevant and related to the question asked in the first place and allowed to pinpoint directly the inconsistent position which will make the process of correcting the inconsistency easy and feasible.

The dialogue respects certain rules and uses predefined locutions like “why”, “understand”, etc. In addition, it makes use of questions and their answers to generate explanations and uses argumentation to weigh different contradicting conclusions. The dialogue also has a turn taking mechanism where the **User** and the **Reasoner** switch turns at each stage. Given this context, the aim of this paper is to propose a new formal model of explanatory dialogues called **EDS** which is used to explain query entailment in inconsistent knowledge bases. We implemented this formal model in a system called **DALEK** (**Di**ALectical **Ex**planation in **Knowledge**-bases). We showed how the use of such model can help in reducing inconsistencies and improve knowledge acquisition in the context of the ANR (French National Agency) funded Dur-Dur project on Durum Wheat sustainability.

In Section 2 we present the *Datalog* \pm logical language used to represent and reason with the knowledge base. Next, in Section 3 we present the extension of the logical language with questions and explanations. Then, in Section 4 we introduce the formal model of explanatory dialogues based on the language of Section 3. After that, in Section 5 & 6 we first present the DALEK framework that implements the proposed formal model, and then the pilot evaluation on knowledge acquisition and inconsistency reduction with explanatory dialogues. Finally, in Section 8 & 9 we discuss related work and conclude the paper.

2. The Logical Language \mathcal{L}

There are two major approaches to represent an ontology: Description Logics (such as \mathcal{EL} , Baader et al. (2005) and DL-Lite, Calvanese et al. (2007) families) and rule-based languages (such as *Datalog* \pm Cali et al. (2009) language, a generalization of Datalog (Ceri et al., 1989) that allows for existentially quantified variables in rules heads). Despite *Datalog* \pm undecidability when answering conjunctive queries, different decidable fragments are studied in the literature (see (Baget et al., 2011b)). 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.

We consider *the positive existential* conjunctive fragment of first-order logic FOL(\exists, \wedge) (Chein & Mugnier, 2009; Baget et al., 2011a). Its language \mathcal{L} is composed of formulas built with the usual quantifiers (\exists, \forall) and *only* the connectors implication (\rightarrow) and conjunction (\wedge).

Vocabulary. We consider first-order vocabularies with constants but no other function symbol. A vocabulary is a pair $\mathcal{V} = (\mathcal{P}, \mathcal{C})$, where \mathcal{P} is a finite set of predicates and \mathcal{C} is a possibly infinite set of constants. A term t over \mathcal{V} is a constant or a variable, different constants represent different values (unique name assumption). We use uppercase letters for constants and lowercase letters for variables.

Atomic formulae. An **atomic formula** (or atom) over \mathcal{V} is of the form $p(t_1, \dots, t_n)$ where $p \in \mathcal{P}$ is an n-ary predicate, and t_1, \dots, t_n are terms. A **ground atom** is an atom with no variables (e.g. $p(A, C)$). A conjunction of atoms is called a **conjunct**. A conjunction of **ground atoms** is called a **ground conjunct**. By convention a ground atom is a ground conjunct. A variable in a formula is free if it is not in the scope of any quantifier. A formula is **closed** if it has no free variables (also known as **sentence**).

Example 2 (Atoms and conjuncts). *As an example, consider the set of constants $\mathcal{C} = \{John\}$, $\mathcal{P} = \{student, teacher, teaches\}$ and a countably infinite set of variables $\mathcal{X} = \{x_1, x_2, x_3, \dots\}$. Then, $teaches(John, x_1)$ is an atom, $teacher(John)$ is a ground atom, $teaches(John, x_1) \wedge teacher(John)$ is a conjunct and $teacher(John) \wedge teaches(John, Tom) \wedge student(Tom)$ is a ground conjunct.*

Facts. Classically, a fact is a ground atom. Baget et al. (2011a) extended this notion, so that a fact may contain existentially quantified variables and not only constants. Thus, a **fact** on \mathcal{V} is the existential closure of a conjunction of atoms over \mathcal{V} . For instance, $F = \exists x_1(teacher(John) \wedge student(x_1) \wedge teaches(John, x_1))$ is an example of a fact where x_1 is an existentially quantified variable. We may omit quantifiers in facts as there is no ambiguity (they are all existentially quantified). As one may notice here, the existential variable permit to represent unknown values which is an interesting property in this language. We denote by $terms(F)$ (resp. $vars(F)$) the set of terms (resp. variables) that occur in F . We exclude duplicate atoms in facts, which allows to see a fact as a set of atoms. For instance, the fact $F = \exists x \exists y(r(x) \wedge p(A, y) \wedge r(x))$ can be seen as $\{p(A, y), r(x)\}$ where $vars(F) = \{x, y\}$ and $terms(F) = \{A\}$. From now on we may use the set notation and the logical notation interchangeably.

Rules and negative constraints. We denote by \vec{x} a vector of variables. An *existential rule* (or simply a rule) is a closed formula of the form $R = \forall \vec{x} \forall \vec{y} (B \rightarrow \exists \vec{z} H)$, where B and H are conjuncts, with $vars(B) = \vec{x} \cup \vec{y}$, and $vars(H) = \vec{x} \cup \vec{z}$.

The variables \vec{z} are called the existential variables of the rule R . B and H are respectively called the *body* and the *head* of R . We denote them respectively $body(R)$ for B and $head(R)$ for H . We may sometimes omit quantifiers and write $R = B \rightarrow H$. As an example of rules we have $R = \forall x \forall y p(x, y) \rightarrow \exists z q(y, z)$. A *negative constraint* (or simply a constraint) is a rule of the form $N = \forall \vec{x} (B \rightarrow \perp)$. For instance, $\forall x p(x) \wedge q(x) \rightarrow \perp$ is a negative constraint that says that x cannot be p and q .

Substitution and homomorphism. Given a set of variables \mathcal{X} and a set of terms \mathcal{T} , a **substitution** σ of \mathcal{X} by \mathcal{T} (notation $\sigma : \mathcal{X} \rightarrow \mathcal{T}$) is a mapping from \mathcal{X} to \mathcal{T} . Given a fact F , $\sigma(F)$ denotes the fact obtained from F by replacing each occurrence of $x \in \mathcal{X} \cap vars(F)$ by $\sigma(x)$. A **homomorphism** from a fact F to a fact F' is a substitution σ of $vars(F)$ by (a subset of) $terms(F')$ such that $\sigma(F) \subseteq F'$ (Baget et al., 2011a).

Example 3 (Homomorphism). Let $F = \{q(A, x)\}$ and $F' = \{q(A, B), r(A)\}$ where $vars(F) = \{x\}$ and $terms(F') = \{A, B\}$. We have two possible substitutions $\sigma_1 = \{(x, A)\}$ and $\sigma_2 = \{(x, B)\}$ where x is substituted by A in σ_1 and by B in σ_2 . When we apply σ_1 (resp. σ_2) on F we get $\sigma_1(F) = \{q(A, A)\}$ (resp. $\sigma_2(F) = \{q(A, B)\}$). It is clear that the substitution σ_2 is a homomorphism from F to F' (unlike σ_1) because $\sigma_2(F) \subseteq F'$ such that $\sigma_2(F) = \{q(A, B)\}$.

Rule Application. A rule $R = B \rightarrow H$ is **applicable** to a fact F if there is a homomorphism σ from B to F . The *application of R to F w.r.t. σ* produces a fact $\alpha(F, R, \sigma) = F \cup \sigma(safe(H))$, where $safe(H)$ is obtained from H by replacing existential variables with fresh variables (not used variables). $\alpha(F, R, \sigma)$ is said to be an **immediate derivation** from F . For instance, let $R = q(x, y) \rightarrow p(x, y)$ and $F = \{q(A, B), r(A)\}$, R is applicable to F because there is a homomorphism from $\{q(x, y)\}$ to $\{q(A, B), r(A)\}$ that substitutes x by A and y by B . The immediate derivation from F is the fact $F' = \{q(A, B), r(A)\} \cup \{p(A, B)\}$.

Derivation and closure. Let F be a fact and \mathcal{R} be a set of rules. A fact F' is called an **\mathcal{R} -derivation** of F if there is a finite sequence (called the **derivation sequence**) $\langle F_0 = F, \dots, F_n = F' \rangle$ such that for all $0 \leq i < n$ there is a rule R which is applicable to F_i and F_{i+1} is an immediate derivation from F_i . Given a fact F and a set of rules \mathcal{R} , the **chase** (or saturation) procedure starts from F and performs rule applications in a breadth-first manner. The chase computes the **closure** of F , i.e. $\text{Cl}_{\mathcal{R}}(F)$, which is the smallest set that contains F and that is closed under \mathcal{R} -derivation, i.e. for every \mathcal{R} -derivation F' of F we have $F' \in \text{Cl}_{\mathcal{R}}(F)$. Many variants of chase procedures have been studied in the literature see Baget et al. (2011a). Given a chase variant C , we call C -finite the class of set of rules \mathcal{R} , such that the C -chase halts on any fact F , consequently produces a finite $\text{Cl}_{\mathcal{R}}(F)$. We limit our work in this paper to these kind of classes.

Entailment. Let F and F' be two facts. $F \models F'$ if and only if there is a homomorphism from F' to F . For instance $\{r(A, x), d(A)\} \models r(A, B)$. Given two facts F and F' and a set of rules \mathcal{R} we say $F, \mathcal{R} \models F'$ if and only if $\text{Cl}_{\mathcal{R}}(F) \models F'$ where \models is the classical first-order entailment (Mugnier, 2011).

Example 4. Let $F = \{q(A, B), r(D), p(x_1, C)\}$ and $\mathcal{R} = \{R_1, R_2\}$ such that $R_1 = q(x_1, y_1) \wedge r(z_1) \rightarrow d(x_1, z_1)$ and $R_2 = p(x_2, y_2) \wedge r(z_2) \rightarrow m(z_2, x_2)$. The following is a derivation sequence: $\langle F_0, F_1, F_2 \rangle$ where $F_0 = F$, $F_1 = \{q(A, B), r(D), d(A, D), p(x_1, C)\}$ and $F_2 = F_1 \cup \{m(D, x_1)\}$. We get F_1 by applying R_1 on F then we get F_2 by applying R_2 on F_1 . We say F_2 is an \mathcal{R} -derivation of F . The closure of F is $\text{Cl}_{\mathcal{R}}(F) = F \cup \{d(A, D), m(D, x_1)\}$.

Knowledge base and inconsistency. Let us denote by \mathcal{L} the language described so far, A knowledge base \mathcal{K} is a finite subset of \mathcal{L} . Precisely, \mathcal{K} is a tuple $(\mathcal{F}, \mathcal{R}, \mathcal{N})$ of a finite set of facts \mathcal{F} , rules \mathcal{R} and constraints \mathcal{N} . Saying that $\mathcal{K} \models F$ means $\text{Cl}_{\mathcal{R}}(\mathcal{F}) \models F$. We say a set of facts \mathcal{F} is **inconsistent** with respect to a set of constraints \mathcal{N} and rules \mathcal{R} if and only if there exists $N \in \mathcal{N}$ such that $\text{Cl}_{\mathcal{R}}(\mathcal{F}) \models \text{body}(N)$. A knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is said to be

inconsistent with respect to \mathcal{R} and \mathcal{N} (inconsistent for short) if \mathcal{F} is inconsistent. We may use the notation $\text{Cl}_{\mathcal{R}}(\mathcal{F}) \models \perp$ to mean the same thing. A conflict in \mathcal{K} is a minimal (w.r.t set inclusion) set of facts in \mathcal{F} that is inconsistent.

Example 5 (Knowledge base). *The following is an example of a knowledge base.*

- $\mathcal{F} = \{teacher(Linda), teaches(John, Tom), teaches(John, Ahmed), teaches(John, x_1), taught_in(Tom, C101), taught_in(Ahmed, S101)\}$
- $\mathcal{R} = \{\forall x_1 \forall x_2 (teaches(x_1, x_2) \rightarrow teacher(x_1) \wedge student(x_2)), \forall x_1 \forall x_2 \forall x_3 (teaches(x_1, x_2) \wedge taught_in(x_2, x_3) \rightarrow teaches_in(x_1, x_3))\}$
- $\mathcal{N} = \{\forall x_1 (teacher(x_1) \wedge student(x_1) \rightarrow \perp)\}$

One particular thing needs clarification. The fact $teaches(John, x_1)$ expresses that there exists an individual x_1 that John teaches, this fact represents an incomplete piece of knowledge. The existential variable x_1 could be a *Oliver* or even *Linda* but all what we know currently is John teaches another one.

In the next section we extend \mathcal{L} to incorporate the notion of questions and explanations.

3. The Language \mathcal{L}^+ : Questions and Explanations

The posing of questions is crucial for the usability of knowledge bases. It is often used when querying and inspecting the content of the knowledge base for information. Questions can take different forms depending on the intention of asking them. In this section we consider the following four types of questions:

- Whether questions.
- Which and who questions.
- Why questions.

The first question is rather simple but yet interesting, it corresponds to the case where one would like to know from a set of choices which one is correct. The second type of questions is the one which is often used in knowledge-base systems where the user asks for certain answers that satisfy certain properties. The third question is rarely used, by posing such question the user asks for explanatory knowledge about a fact which holds true in the knowledge base. This list of questions is far from being exhaustive but we believe that they are expressive enough to capture more than usual knowledge-base capabilities.

To formally integrate these questions, we extend the language \mathcal{L} to \mathcal{L}^+ by incorporating the language of the logic of questions (erotetic³ logic, EL for short) proposed by Wisniewski (2013). Note that other logics can be considered, we limit ourselves to the EL for its simplicity. When questions are answered, explanations are used to explain why certain answers hold for certain questions. Since EL does not account for explanations, we incorporate a new type of questions in \mathcal{L}^+ called why questions.

Following Hamblin (1958), questions are represented as a set of their possible answers, this is referred to as the set-of-answers methodology (SAM). It has been advocated in Hamblin (1958) postulates: (1) An answer to a question is a statement, (2) knowing what counts as an answer is equivalent to knowing the question and (3) The possible answers to a question are an exhaustive set of mutually exclusive possibilities.

Given the language \mathcal{L} , the erotetic language \mathcal{L}^+ is the extension of \mathcal{L} 's vocabulary by the following erotetic signs: question mark (?), curly brackets ($\{, \}$), **S**, **U**, **W**, and the comma.

Now let us define the formation rules for a question in \mathcal{L}^+ . The simplest kind of questions is called a question of the first kind.

Definition 1 (Question of the first kind). *A **question of the first kind** in \mathcal{L}^+ is an expression of the form $?\{A_1, \dots, A_n\}$ where $n \geq 1$ and A_1, \dots, A_n*

³From Greek which means *question*.

are syntactically different ground conjuncts. If $Q = ?\{A_1, \dots, A_n\}$ is a question then A_1, \dots, A_n are its direct answers.

Note that the brackets $(\{, \})$ belong to the vocabulary of the language \mathcal{L}^+ and they do not refer to a set-theoretic notation. Questions are object-level expressions of a strictly defined form, thus $?\{A_1, A_2\}$ and $?\{A_2, A_1\}$ are syntactically two different questions. Nevertheless, in the metalanguage level we refer to the set of direct answers to a question Q as dQ . Note that dQ is a finite set that contains at least two elements. Questions of the first kind correspond to whether-questions and they are read as: “Is it the case that A_1 , or is it the case that A_2, \dots , or is it the case that A_n ?”.

Example 6. Consider the knowledge base of Example 5 the following is a question of the first kind: $Q = ?\{teacher(John), teacher(Tom)\}$.

Example 7. Let us imagine a group of three friends, James, Carla and John, who want to play football. A question of the first kind would be: “ $Q =$ Who has the Ball: James, Carla or John?”. One direct answer is James has the ball. Formally, given the ground atoms $A_1 = has(James, Ball)$, $A_2 = has(Carla, Ball)$ and $A_3 = has(John, Ball)$, the question is put as: $Q = ?\{A_1, A_2, A_3\}$.

Any set of ground conjuncts defines a question (of the first kind) in \mathcal{L}^+ . Note that we do not consider “James and John have the ball”? nor “James has not the ball” as direct answers because they are partial answers. Direct answers are those which give the right amount of information to answer the question, no more and no less.

A question of the first kind cannot express questions of the form “Which \vec{x} , is such that $A(\vec{x})$?” where A is an existentially quantified conjunct with \vec{x} as its free variables. In this case the question is asking about an individual or a n-tuple of individuals that satisfy A (asking for one example). In short, the difference is that here we ask for unknown individuals using free variables.

Questions of the second kind encompass existential and open question.

Definition 2 (Existential questions). An *existential question* in \mathcal{L}^+ is an expression of the form $?S(A(\vec{x}))$ such that $A(\vec{x})$ is existentially quantified conjunct with \vec{x} as its free variables and $|\vec{x}| \geq 1$. A direct answer to $?S(A(\vec{x}))$ follows the schema $A(\vec{x}/\vec{U})$ such that $\vec{x} = (x_1, \dots, x_n)$ and $\vec{U} = (U_1, \dots, U_n)$ where for all $i \in \{1, \dots, n\}$, \vec{x}/\vec{U} means that the variable x_i is substituted by the constant U_i . The set of its direct answers is $dQ_S = \{A(\vec{x}/\vec{U}_1), \dots, A(\vec{x}/\vec{U}_k)\}$, $k > 1$.

Thus existential questions consists of the erotetic constants $?$ and S followed by an existentially quantified conjunct enclosed in parentheses.

Example 8. Let \mathcal{K} be the KB of the Example 5. Consider the following existential question $Q = ?S(A(x, y))$ such that $A(x, y) = \text{teacher}(\text{John}) \wedge \text{teaches}(\text{John}, x) \wedge \text{teaches.in}(\text{John}, y)$. This question asks which student and class such that James teaches that student in that class. Its set of direct answers is dQ hereafter, note that for space reasons dQ is abbreviated with the following notation:

$dQ_S = \{A(x, y) | x, y \in \mathcal{C}\}$ such that $\mathcal{C} = \{\text{Ahmed}, \text{Linda}, \text{John}, \text{Tom}, \text{S101}, \text{C101}\}$ is the set of all the constants that appear in \mathcal{K} .

Note that $dQ_S \cup \{A(\text{Tom}, \text{M120})\}$ is not a correct set of direct answers because M120 is not a constant that appears in \mathcal{K} .

This type of questions fails also to express the question “What are some \vec{x} such that $A(\vec{x})$?”. They only give one possible answer. The answer $A(\text{Tom}, \text{C101}) \wedge A(\text{Ahmed}, \text{S101})$ is not considered here although it is correct.

In what follows we introduce the type of open questions.

Definition 3 (Open questions). An *open question* is an expression of the form $?O(A(\vec{x}))$ such that $A(\vec{x})$ is an existentially quantified conjunct with free variables \vec{x} and $|\vec{x}| \geq 1$. Let $?S(A(\vec{x}))$ be its existential question and dQ_S its direct answers. A direct answer to $?O(A(\vec{x}))$ follows the schema:

$A(\vec{x}/\vec{U}_1) \wedge \dots \wedge A(\vec{x}/\vec{U}_k)$, $k > 1$ where $A(\vec{x}/\vec{U}_1), \dots, A(\vec{x}/\vec{U}_k)$ are the elements of dQ_S .

The set of direct answers $dQ_{\mathcal{O}}$ to an open question $?O(A(\vec{x}))$ consists of all elements from the set $dQ_{\mathcal{S}}$ of its existential question and all of their conjunctions.

Example 9. The following is an open questions $Q = ?O(A(x, y))$ such that $A(x, y) = \text{teacher}(\text{James}) \wedge \text{teaches}(\text{James}, x) \wedge \text{teaches_in}(\text{James}, y)$. The set of its direct answers is:

$$dQ_{\mathcal{O}} = dQ_{\mathcal{S}} \cup \{A(\text{Tom}, C101) \wedge A(\text{Ahmed}, S101), A(\text{Linda}, S101) \wedge A(\text{John}, S101) \wedge A(\text{Tom}, S101), \dots\}.$$

Still, the framework defined so far does not account for “Why” questions. For instance, the question “Why is the case that A ?” cannot be neither formalized nor answered. Let us define the syntax of such questions and its direct answers.

Definition 4 (Why questions). A **why question** is an expression of the form $?W(A)$ such that A is an existentially quantified conjunct with no free variables (i.e. a fact) called the **explanandum**. Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, a direct answer to $?W(A)$ with respect to \mathcal{K} is a set of facts and rules $\mathcal{E} \subseteq \mathcal{F} \cup \mathcal{R}$ called an **explanation** such that:

1. neither $\mathcal{E} \models \perp$ nor $\mathcal{E} \cup \{A\} \models \perp$ (consistency).
2. $\mathcal{E} \models A$ (entailment).
3. there is no $\mathcal{E}' \subset \mathcal{E}$ that verifies the condition (1) and (2) (minimality).

The set of all direct answers $dQ_{\mathcal{W}}$ is the set of all possible explanations. $\text{facts}(\mathcal{E})$ denotes the factual part of the explanation \mathcal{E} and $\text{explanandum}(\mathcal{E})$ is the object of the explanation, i.e. A .

A direct answer to a why question symbolizes an explanation for the explanandum. An explanation \mathcal{E} is a set of facts and rules that necessarily entail the explanandum. The explanation should be consistent in itself and with the explanandum and the explanation should be minimally relevant, i.e. contains only information that contributes to the entailment of the explanandum. Note

that the explanandum is a fact and does not contain free variables, this position is advocated in Hempel & Oppenheim (1948); Walton (2011) where the explanandum should be of a factual nature.

Example 10. *Imagine that we have a knowledge base about employee and their salaries. The following is a why question $Q = ?\mathbf{W}(A)$ such that $A = \exists x \text{has_salary}(\text{Tom}, x)$ which can be read as “Why Tom has a salary” or “Why is it the case that Tom has a salary”. A direct answer to Q is:*

$$\mathcal{E} = \{\text{works_at}(\text{Tom}, \text{UM}), \text{university}(\text{UM}), \text{works_at}(x_1, y_1) \wedge \text{university}(y_1) \rightarrow \text{has_salary}(x_1, z_1)\}. \text{ Where}$$

$$\text{facts}(\mathcal{E}) = \{\text{works_at}(\text{Tom}, \text{UM}), \text{university}(\text{UM}), \text{works_at}(x_1, y_1)\}.$$

Another direct answer could be:

$$\mathcal{E}' = \{\text{retired_from}(\text{Tom}, \text{UM}), \text{university}(\text{UM}), \text{retired_from}(x_2, y_2) \wedge \text{university}(y_2) \rightarrow \text{has_salary}(x_2, z_2)\}.$$

Note that the following:

$$\mathcal{E}'' = \{\text{works_at}(\text{Tom}, \text{UM}), \text{university}(\text{UM}), \text{student}(\text{Ahmed}), \text{works_at}(x_3, y_3) \wedge \text{university}(y_3) \rightarrow \text{has_salary}(x_3, z_3)\}$$

is not an explanation because *student(Ahmed)* is irrelevant. Note also that, $\mathcal{E} \setminus \{\text{works_at}(\text{Tom}, \text{UM})\}$ is not an explanation because it violates entailment.

We draw the intention of the reader that the question $Q' = ?\mathbf{W}(A)$ such that $A = \text{has_salary}(\text{Tom}, 17000)$ has another reading different than Q . It is read as “Why Tom has a salary 17000?”, which would have a different explanation.

From now on, unless otherwise stated, we make now distinction between the different kinds of questions and we call them questions for short and we refer by dQ to their set of direct answers. Note that we say two questions are equal if and only if they have the same set of direct answers.

The semantics of \mathcal{L}^+ is called the **M**inimal **E**rotetic **S**emantics (MiES) (Wisniewski, 2013) which is the usual model-theoretic semantics of first-order logic augmented with a new concept called soundness of questions. MiES does not regard questions as are true or false but rather sound or unsound.

Definition 5 (Soundness). *Given a knowledge base \mathcal{K} , a question Q is sound relative to \mathcal{K} if and only if $\mathcal{K} \models dQ$.*

This means that a sound question relative to a knowledge base \mathcal{K} is a question that at least one of its direct answers is entailed by \mathcal{K} .

Example 11. *The question $Q = \{teacher(John), teacher(Tom)\}$ of Example 6 is sound because $\mathcal{K} \models teacher(John)$.*

In human dialogues, questions are evoked naturally when some body of information becomes available within the discussion. Sometimes questions are not allowed to be asked if they are based on false information. These two concepts correspond to *evocation* and *presupposition* in the logic of questions.

Definition 6 (Presupposition). *A fact F is a presupposition of a question Q if and only if for each $A \in dQ$, $A \models F$. The set of all presuppositions of a question Q is denoted by $\text{pres}(Q)$. When $\text{pres}(Q) \neq \emptyset$ we call Q a regular question.*

Example 12. *One presupposition of the question $Q = \{teacher(John), teacher(Tom)\}$ is $\exists x teacher(x)$ because $teacher(John) \models \exists x teacher(x)$ and $teacher(Tom) \models \exists x teacher(x)$. Informally, the question Q presupposes that there is a teacher.*

Information can give raise to questions, these questions are usually asked to complete our knowledge about the presented information. We define hereafter evocation between a fact and a question.

Definition 7 (Evocation). *Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$. A fact F evokes a question Q in \mathcal{K} iff:*

1. $\forall A \in dQ, F \not\models A$ (informativeness), and,
2. $\mathcal{K} \models dQ$ (soundness relative to \mathcal{K}).

An evoked question with respect to a fact F should be informative relative to F , that means we cannot answer Q by any direct answer that can be inferred from F only. The second condition stipulates that Q should sound relative to the knowledge base \mathcal{K} . For instance, in the Example 5 the fact

$F = student(Tom) \wedge teacher(John) \wedge teacher(Linda)$ evokes the question $Q = ?\{teaches(John, Tom), teaches(Linda, Tom)\}$, i.e. “Who teaches Tom: John or Linda?”. It is clear that the first condition is met since $F \not\models teaches(John, Tom)$ and $F \not\models teaches(Linda, Tom)$. Furthermore, the second condition is verified because $\mathcal{K} \models dQ$. In fact, $\mathcal{K} \models teaches(John, Tom)$ which is the correct answer in \mathcal{K} .

Questions imply questions, this is called erotetic implication. For instance,

Definition 8 (Implication). *A question Q implies another question Q' in \mathcal{K} if and only if:*

- if $\mathcal{K} \models dQ$ then $\mathcal{K} \models dQ'$ (soundness relative to \mathcal{K}).
- for each $B \in dQ'$ there exists $\Delta \subset dQ$ such that $\Delta \neq \emptyset$ and $B \models \Delta$.

If the implying question is sound relative to \mathcal{K} then the implied question must be sound relative to \mathcal{K} . Each direct answer to the implied question limits the scope of possible correct answers of the implying question. In other words, it gives a partial answer to the initial question.

Example 13. *Consider the KB of Example 5. Consider the question $Q = ?\{teaches_in(Tom, S101), teaches_in(Ahmed, S101), teaches_in(John, S101), teaches_in(Linda, S101)\}$. This question implies $Q' = ?\{teaches_in(John, S101), teaches_in(Ahmed, S101)\}$. The question $Q'' = ?\{teaches_in(Tom, S101), teaches_in(Ahmed, S101)\}$ is not implied by Q because it is not sound in \mathcal{K} .*

Let us close this section by stating that the set of well-formed formulae of \mathcal{L}^+ is composed of all well-formed formulae of \mathcal{L} and questions of the previous kinds.

4. The Formal System of Explanatory Dialogues EDS

Explanatory dialogue is a two-party dialogue that takes place between an explainer and an explainee (**Reasoner** and **User** respectively in our context). The speech acts of requesting and providing an explanation are represented as

utterances. The goal of the dialogue is to get the explainee to understand an assertion done by the explainer (Walton, 2011).

In this section we describe a dialogue system of explanatory dialogues that is inspired from Walton (2011); Arioua & Croitoru (2015). We start by presenting the dialogue’s syntax (Section 4.1), then the dialogue’s semantics (Section 4.2), i.e. the meaning of utterances within a dialogue. After that, we discuss the integration of commitment and understanding stores to preserve the coherence of the dialogue (Section 4.3). Next, we add argumentation to handle the problem of inconsistency (Section 4.4). Finally, we discuss the conditions of termination and success (Section 4.5) and give a detailed example of explanatory dialogues (Section 4.6).

4.1. The Syntax

Dialogues are composed of two main parts, the syntax and the semantics (McBurney & Parsons, 2009). We follow McBurney et al. (2002) and present the syntax independently from the semantics. We start by defining the general framework over which the explanatory dialogue system is built.

Definition 9 (Dialogue system). *A dialogue system is a tuple $\mathcal{D}_{sys} = (\mathcal{Pr}, \mathcal{C}, \mathbb{R}, \mathcal{L}, \mathcal{K})$ such that $\mathcal{Pr} = \{\mathbf{U}, \mathbf{R}\}$ is the set of participants where \mathbf{U} refers to **User** and \mathbf{R} to **Reasoner**, \mathcal{C} is a finite set of the allowed locutions within the dialogue, \mathbb{R} is an irreflexive binary relation defined over \mathcal{U} called the reply relation, \mathcal{L} is an arbitrary content language and $\mathcal{K} \subseteq \mathcal{L}$ is a background knowledge base accessible by both participants.*

This dialogue system describes the general components of an arbitrary dialogue. In what follows we instantiate these components to describe a dialogue system called *the explanatory dialogue system* (EDS).

Participants. In a general dialogue system the participants refer to two agents (sometimes more) with possibly different nature, they may be human agents or mixed agents (e.g. machine with human) where each participant plays a role. In the explanatory dialogue system EDS we are interested in the case of

machine-human agents where the machine is called the **Reasoner** and denoted as **R**, while the human is called **User** and denoted as **U**. **U** plays the role of the *explainee* and the other plays the role of an *explainer*.

Topic and content language. Following Walton (2011); Arioua & Croitoru (2015) the topic of any dialogue of EDS is a discussion that aims to get **U** to understand why a query Q is entailed from \mathcal{K} . In this dialogue system, **R** tries to provide explanations to **U**. The content language defines the language by which the participants exchange information. Here we consider the *formal* language \mathcal{L}^+ previously seen in Section 3.

Background knowledge. The background knowledge is the knowledge mainly held by the **Reasoner** and accessible to the **User**. It is denoted as $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and it contains a set of facts, rules and constraints. For the present, the knowledge base \mathcal{K} is assumed to be consistent⁴.

In what follows we detail the rest of the components, i.e. locutions and the reply relation.

One of the important component here that distinguish EDS from other dialogue systems ⁵ is the set of allowed locutions and their possible replies.

In Table 1 column “Locutions” the allowed locutions within EDS are presented. These locutions can be uttered in the dialogue. The set of locutions is neither disjoint nor equal, the difference is due to the asymmetry of roles where the **Reasoner** plays the role of an explainer whereas the **User** plays the role of an explainee. For instance, the **User** is the one who asks for an explanation (EXPLAIN) and the **Reasoner** is one who provides the explanation (ATTEMPT) not the other way around. Nevertheless, this asymmetry does not prevent both parties from using questions within the dialogue for different purposes (ASK and ANSWER). For instance, the **Reasoner** can use questions to prob the **User**’s

⁴In Section 4.4 we handle the case where the background knowledge is inconsistent.

⁵For instance, the dialogue systems of argumentation CB (Walton, 1984), DC (Mackenzie, 1979), PPD (Walton & Krabbe, 1995), to name a few.

knowledge in order to adjust the level of details in the explanation. The **User** can ask questions also as follow-ups to deepen his/her understanding and knowledge.

The reply relation \mathbb{R} in Table 1 specifies which locution replies to which locutions. As it is indicated, the EXPLAIN request locution is either replied to by an ATTEMPT that provides an explanation, or by a declaration of inability INABILITY. ATTEMPT is replied to by an explanation request EXPLAIN which asks for an explanation of some parts of the first explanation. Or by NEGATIVE to dis-acknowledge understanding or by POSITIVE to acknowledge understanding. A question (ASK) can either be replied to by a question (ASK) or an answer (ANSWER). The locution ANSWER can be replied to by either a question locution ASK or by an explanation request EXPLAIN. As in day-to-day dialogues, a question can imply another question, so the latter can be introduced as a consequence. Answers can also evoke questions. These correspond respectively to the concept of implication and evocation discussed in Section 3, they will be further elaborated in Subsection 4.2 (semantics).

Generally, a dialogue is a sequence of **utterances** between two parties (or more). We follow Atkinson et al. (2005) and we represent utterances by a two-layer syntax: the wrapper layer and the content layer. The wrapper layer encompasses locutions which represent the illocutionary force of the inner content, e.g. EXPLAIN, ATTEMPT, etc. The content layer includes the following components: the identifier x of the speaker, the identifier i of the utterance, the target t of the utterance and the content A of the utterance expressed in the content language \mathcal{L}^+ .

Definition 10 (Utterance). *An utterance u has the form $X(x, i, t, A)$ such that: $t, i > 0$, $x \in \mathcal{P}r$, A is a well-formed formula of \mathcal{L}^+ and X is a locution which is in one of the following sets:*

- $\mathcal{C}_R = \{\text{ATTEMPT, INABILITY}\}$ (*R's allowed locutions*).
- $\mathcal{C}_U = \{\text{EXPLAIN, POSITIVE, NEGATIVE}\}$ (*U's allowed locutions*).

Locutions	R	U	Description	Reply
EXPLAIN		√	U requests an explanation	ATTEMPT or INABILITY
ATTEMPT	√		R provides an explanation	NEGATIVE, or POSITIVE, or ASK, or EXPLAIN
ASK	√	√	R or U asks a question	ANSWER or ASK or ANSWER(\emptyset)
ANSWER	√	√	R or U answers a question	ASK or EXPLAIN
ANSWER(\emptyset)	√	√	R or U has no answer to a question	no reply
POSITIVE		√	U acknowledges understanding or partial understanding	no reply
NEGATIVE		√	U disacknowledges understanding	no reply
INABILITY	√		R declares inability to explain	no reply

Table 1: Locutions and replies.

- $\mathcal{C}_U = \{\text{ASK}, \text{ANSWER}\}$ (allowed locutions for both R and U).

Notation 1. Let $u = X(x, i, t, A)$ be an utterance. We use the following notations $\text{loc}(u) = X$, $\text{part}(u) = x$, $\text{id}(u) = i$, $\text{target}(u) = t$ and $\text{content}(u) = A$ to denote respectively the locution, the participant, the identifier, the target and the content of the utterance u .

Not all the utterances are syntactically valid. In what follows we define when a given utterance is syntactically valid. We also define syntactical equality between utterances.

Definition 11 (Syntactical validity and equality). Let u be an utterance.

- We say u is **syntactically valid** if and only if (1) $\text{id}(u) \geq 1$, (2) $\text{target}(u) > \text{id}(u)$ and $\text{loc}(u) \in \mathcal{C}_{\text{part}(u)}$. If $\text{loc}(u) = \text{EXPLAIN}$ or $\text{loc}(u) = \text{ASK}$ then $\text{target}(u)$ may be set to 0.
- We say u is **syntactically equal** to another utterance u' and we write

$u = u'$ if and only if $\text{loc}(u) = \text{loc}(u')$, $\text{part}(u) = \text{part}(u')$, $\text{target}(u) = \text{target}(u')$ and $\text{content}(u) = \text{content}(u')$.

The syntactical validity of utterances dictates that the locution of any utterance has to be in the set of x 's allowed locutions and the target identifier should be greater than the identifier of the utterance it self except for the utterance `EXPLAIN()` and `ASK()`. An utterance is syntactically equal to another if it has the same locution, participant, target and content. For notation purposes if the speaker x is \mathbf{R} then the recipient \bar{x} is \mathbf{U} and vice versa. We sometimes omit the parameters of the utterance and we write only the locution followed by a closed parentheses, e.g. `ATTEMPT()`.

After defining the most important part of a dialogue, i.e. utterances, we define a dialogue as a possibly infinite sequence of syntactically valid utterances.

Definition 12 (Explanatory dialogue). *An explanatory dialogue (dialogue for short) D is a possibly infinite sequence $D_n = \langle u_1, u_2, \dots, u_n, \dots \rangle$ of syntactically valid utterances u_i where $i > 0$ and for all $u_i \in D$, $\text{id}(u_i) = i$. D_i is the dialogue where u_i is the most recent utterance, D_i is referred to as the dialogue at stage i . D_0 is the empty dialogue. The set of all possible dialogues is denoted by $\mathcal{D}^{<\infty}$.*

The definition imposes one constraint, i.e. the utterances within the dialogue should be syntactically valid.

Example 14 (Dialogue). *Consider the following dialogue of 4 steps:*

$D_4 = \langle \text{EXPLAIN}(\mathbf{U}, 1, 0, Q), \text{ATTEMPT}(\mathbf{R}, 2, 1, \mathcal{E}), \text{NEGATIVE}(\mathbf{U}, 3, 2, A), \text{POSITIVE}(\mathbf{U}, 4, 2, \mathcal{E}) \rangle$.

This is a dialogue where \mathbf{U} asks for an explanation. Next, \mathbf{R} fulfills the request. After that \mathbf{U} disacknowledges understanding then finally acknowledges understanding.

The parameters of the utterance contextualize the intention of uttering such locution. Table 2 indicates for each utterance its complete parameters.

As in the dialogue of Example 1, \mathbf{R} and \mathbf{U} take turns. The most basic form of turn taking is unique-utterance turn taking. This turn taking gives the par-

Utterance	Meaning
$\text{EXPLAIN}(\mathbf{U}, i, t, Q)$	\mathbf{U} asks for an explanation of A such that $Q = ?\mathbf{W}(A)$ is a why question
$\text{ATTEMPT}(\mathbf{R}, i, t, \mathcal{E})$	\mathbf{R} responds to \mathbf{U} 's t^{th} utterance by an explanation \mathcal{E} .
$\text{ASK}(x, i, t, Q)$	x asks \bar{x} a question $Q = ?\mathbf{O}(A)$ or $Q = ?\mathbf{S}(A)$ or $Q = ?\{A_1, \dots, A_n\}$. If $t > 0$, then x responds to the utterance of $\text{id} = t$ by a another question Q .
$\text{ANSWER}(x, i, t, A)$	x answers to \bar{x} 's t^{th} utterance by A .
$\text{ANSWER}(x, i, t, \emptyset)$	x has no answer to \bar{x} 's t^{th} utterance.
$\text{NEGATIVE}(\mathbf{U}, i, t, A)$	\mathbf{U} has not understood the explanation A (or part of it) of the utterance with $\text{id} = t$.
$\text{POSITIVE}(\mathbf{U}, i, t, A)$	\mathbf{U} has understood the explanation A (or part of it) of the utterance with $\text{id} = t$.
$\text{INABILITY}(\mathbf{R}, i, t, Q)$	\mathbf{R} is unable to provide an explanation for the explanation request of utterance with $\text{id} = t$.

Table 2: x indicates \mathbf{U} and \mathbf{R} . If the participant x is \mathbf{R} then the recipient \bar{x} is \mathbf{U} and vice versa. Note that A, Q, \mathcal{E} are well-formed formulae of \mathcal{L}^+ , the semantics is not considered here.

ticipants the possibility to advance one utterance then hand out the turn to the other to respond. A more liberal turn taking is the multi-utterance turn taking where the turn shifts after several utterances (which is the case for the example). As defined in Prakken (2006) we choose the multi-utterance turn taking for its generality. Sure, this turn taking makes the dialogue difficult to handle computationally, however it offers a high flexibility and a natural correspondence with day-to-day explanatory dialogues where one uses sufficient utterances to express his/her point.

Definition 13 (Turn taking function). *A turn taking function \mathcal{T} is defined over the set of all possible dialogues as follows: $\mathcal{T} : \mathcal{D}^{<\infty} \rightarrow 2^{\{\mathbf{U}, \mathbf{R}\}}$. \mathcal{T} assigns to every dialogue the next legal turn as follows:*

- $\mathcal{T}(D_0) = \{\mathbf{U}\}$, $\mathcal{T}(D_1) = \{\mathbf{R}\}$, else $\mathcal{T}(D_i) = \{\mathbf{U}, \mathbf{R}\}$, $\forall i > 0$.

Example 15 (Count'd Example 1). $\mathcal{T}(D_0) = \{\mathbf{R}\}$, $\mathcal{T}(D_1) = \{\mathbf{U}\}$, $\mathcal{T}(D_6) = \{\mathbf{U}\}$.

The syntactical validity of utterances is context-independent and it cannot serve alone as a basis to define a syntactical valid dialogue. We need to look at the syntactical validity of replies also.

Definition 14 (Syntactically valid reply). *Given a finite dialogue $D_n = \langle u_1, u_2, \dots, u_n \rangle$ at stage n . We say the reply u_i to u_j is syntactically valid in D_n if and only if :*

- (1) $(\text{loc}(u_i), \text{loc}(u_j)) \in \mathbb{R}$.
- (2) $\text{part}(u_i) = \mathcal{T}(D_{i-1})$.
- (3) $\text{id}(u_i) = \text{id}(u_{i-1}) + 1$.
- (4) $\text{target}(u_i) = \text{id}(u_j)$.
- (5) $\text{part}(u_i) \neq \text{part}(u_j)$.

If none of the previous conditions are met then the reply u_i to u_j is syntactically invalid in D_n .

The clauses impose that the locution of u_i is a correct reply with respect to the reply relation \mathbb{R} and it is the turn of the participant $\mathbf{part}(u_i)$ to speak. In addition, the identifier should be incremental, and the target utterance has to be uttered sometime in the past and the participant $\mathbf{part}(u_i)$ should not reply to himself.

Example 16 (Reply). *Consider the following dialogue:*

$D_8 = \langle \text{EXPLAIN}(\mathbf{U}, 1, 0, Q_1), \text{ATTEMPT}(\mathbf{R}, 2, 1, \mathcal{E}_1), \text{ANSWER}(\mathbf{R}, 3, 1, B),$
 $\text{ASK}(\mathbf{R}, 4, 0, Q_2), \text{ASK}(\mathbf{U}, 5, 4, Q_3), \text{ANSWER}(\mathbf{R}, 6, 4, C), \text{ATTEMPT}(\mathbf{R}, 7, 1, \mathcal{E}_1),$
 $\text{POSITIVE}(\mathbf{U}, 8, 2, \mathcal{E}_1) \rangle$.

The reply $\text{ANSWER}(\mathbf{R}, 3, 1, B)$ to $\text{EXPLAIN}(\mathbf{U}, 1, 0, Q_1)$ is syntactically invalid because it is not in \mathbb{R} as opposed to the reply $\text{ASK}(\mathbf{U}, 5, 4, Q_3)$ to $\text{ASK}(\mathbf{R}, 4, 0, Q_2)$ which is syntactically valid. Notice also that, the reply $\text{ANSWER}(\mathbf{R}, 6, 4, C)$ is syntactically invalid because \mathbf{R} is replying to itself (i.e. replies to $\text{ASK}(\mathbf{R}, 4, 0, Q_2)$).

As one may notice, the reply of $\text{ATTEMPT}(\mathbf{R}, 7, 1, \mathcal{E}_1)$ to $\text{EXPLAIN}(\mathbf{U}, 1, 0, Q)$ is considered syntactically valid with respect to the conditions seen before. The problem is that this reply is a duplicate of another reply, i.e. $\text{ATTEMPT}(\mathbf{R}, 2, 1, \mathcal{E}_1)$ to $\text{EXPLAIN}(\mathbf{U}, 1, 0, Q)$. In fact reply's validity is a concept with a limited scope, it is only concerned with a local context, i.e. between two utterances. In what follows we capture the global context of dialogues and we introduce *syntactical validity of a dialogue*.

Definition 15 (Syntactically valid dialogues). *Given a finite dialogue D_n at stage n such that $n \geq 0$. A dialogue D_n is syntactically valid if it respects the following rules:*

1. **Empty dialogue rules** ($n = 0$):

(R_1) D_0 is legal.

2. **Commencement rules** ($n = 1$):

(R_2) $D_1 = \langle u_1 \rangle$ is syntactically valid iff $u_1 = \text{EXPLAIN}(\mathbf{U}, 1, 0, Q)$ is syntactically valid.

3. *Dialogue rules* ($n \geq 2$):

(R_3) D_{n-1} should be syntactically valid and the reply u_n to u_j where $j = \mathbf{target}(u_n)$ is syntactically valid and there is no utterance u_i , $i < n$ such that $\mathbf{target}(u_i) = \mathbf{target}(u_n)$ and $u_i = u_n$.

The definition indicates that an empty dialogue is a syntactically valid dialogue. Furthermore, a syntactically valid dialogue always starts with an explanation request made by U. It also imposes that a dialogue is syntactically valid all replies are syntactically valid and no utterance is repeated.

Note that termination rules, commitments rules and other rules which follows the schemata of McBurney & Parsons (2009) will be presented in the following section where we handle the semantics and commitments of the dialogue.

4.2. *The Semantics*

We address the semantic aspect of the dialogue where we are concerned with the content of the utterances and the meaning behind a sequence of utterances. For instance, the utterance $\text{EXPLAIN}(U, 1, 0, A)$ is syntactically valid but it would not be semantically valid if A were not to be a semantically sound “why question” (*cf.* Definition 4). The same applies to the utterance $\text{ATTEMPT}(R, 2, 1, \mathcal{E})$ if \mathcal{E} were not a correct answer to the why question asked in the utterance with identifier 1.

As in the previous section, we start by defining semantic validity of utterances and replies, then we define the semantic validity of a dialogue. Furthermore, we introduce more notions (e.g. commitments, understanding store, etc.) that preserve the coherence of the dialogue.

In Table 3 we present the conditions under which a given utterance is considered semantically valid. Questions of $\text{ASK}()$ and $\text{EXPLAIN}()$ should be sound in \mathcal{K} to ensure that they are relevant within the dialogue. In $\text{ASK}()$ the question should be open, existential, or first kind because why questions have their own utterance $\text{EXPLAIN}()$. We make as a condition that if a question of ASK has a target 0 then one of its presuppositions has to be entailed from the background

knowledge. That means a question which does not reply to a previous utterance can only be introduced if its presupposition is established. By doing so, we can maintain the relevance of questions with respect to the main topic of the dialogue. The content of `ANSWER()` should be a ground conjunct because direct answers to open, existential, or first kind questions are always a ground conjunct. The content of `POSITIVE()` is a set of facts because it states which fact that have been understood by `U`. The content of `NEGATIVE()` is an explanation \mathcal{E} where the `User` declares clearly that he/she has not understood \mathcal{E} .

Utterances	Conditions
<code>EXPLAIN(U, i, t, Q)</code>	$Q = ?\mathbf{W}(F)$ is a why question that is sound in \mathcal{K} .
<code>ATTEMPT(R, i, t, \mathcal{E})</code>	\mathcal{E} is an explanation (cf. Definition 4).
<code>ASK(x, i, t, Q)</code>	Q is either an open question (resp. existential or first kind) that is sound in \mathcal{K} . If $t = 0$ then there exists $F \in \text{pres}(Q)$ such that $\mathcal{K} \models F$.
<code>ANSWER(x, i, t, A)</code>	A is a ground conjunct (cf. Section 2).
<code>ANSWER(x, i, t, \emptyset)</code>	no condition.
<code>NEGATIVE(U, i, t, \mathcal{E})</code>	\mathcal{E} is an explanation.
<code>POSITIVE(U, i, t, X)</code>	X is a set of facts or an explanation.
<code>INABILITY(R, i, t, Q)</code>	Q is a why question that is sound in \mathcal{K} .

Table 3: The utterances and their semantic conditions. Recall, \mathcal{K} is the background knowledge base.

The semantic validity should also be considered within a context where replies are taken into account. Table 4 indicates the conditions under which a reply is semantically valid. For instance, a reply by the utterance `ATTEMPT(R, 2, 1, \mathcal{E})` to the utterance `EXPLAIN(U, 1, 0, Q)` in a dialogue is syntactically valid but it would not be semantically valid if A were not to be a direct answer to the why question Q .⁶

The utterance `EXPLAIN(U, j, i, Q)` is used to open the dialogue by the `User`

⁶Recall that a direct answer to a why question is called an explanation.

to ask for an explanation about a fact. If the **Reasoner** cannot explain then it utters $\text{INABILITY}(\mathbf{R}, i, t, Q)$ as a reply. If the **Reasoner** has an explanation then it issues $\text{ATTEMPT}(\mathbf{R}, i, t, \mathcal{E})$ which may be replied to by confirm understanding or partial understanding. The former is done by uttering $\text{POSITIVE}(\mathbf{U}, i, t, X)$ such that X is the explanation provided in the first place, the latter is done by uttering $\text{POSITIVE}(\mathbf{U}, i, t, X)$ where X is a proper subset of the factual part of the explanation in u_i . The other reply is by stating $\text{NEGATIVE}(\mathbf{U}, i, t, \mathcal{E})$. Another interesting reply is $\text{EXPLAIN}(\mathbf{U}, j, i, Q')$ which is another explanation request about the provided explanation. This is similar to the case when one asks “Why the room is dark?” and has received the explanation “because there is no electricity”, one can ask “why there is no electricity?”.

Questions in $\text{ASK}(x, i, t, Q)$ can imply other questions or can be answered by $\text{ANSWER}(\bar{x}, j, i, A)$. Answers can evoke questions or explanation requests as mentioned in the ninth and tenth row. Explanation attempts can also evoke questions as mentioned in the fifth row. If the **Reasoner** somehow were not been able to answer a question then it utters $\text{ANSWER}(\bar{x}, j, i, \emptyset)$.

4.3. Commitment and Understanding Stores

In order to maintain the coherence of the dialogue, the **Reasoner** and the **User** should not contradict themselves. Argumentation dialogues have a long standing tradition in handling this problem by using commitment stores. We follow (Mackenzie, 1979) and equip EDS with commitment stores. Another problem we may encounter are *circular explanations* (Gratton, 1994). They appear when the **Reasoner** tries to explain something with an explanation that is partially not understood. To handle such problem we propose a new store called *understanding store* which is exclusively attributed to the **User**.

Understanding store serves as an understanding indicator of the **User**’s current understanding state. Note that this store represents what is not yet understood instead of what has been understood. The *commitment store* represents the **User**’s (resp. the **Reasoner**) commitments to the truthfulness of certain statements.

Utterances	Replies	Conditions
EXPLAIN($\mathbf{U}, i, 0, Q$)	ATTEMPT($\mathbf{R}, j, i, \mathcal{E}$)	\mathcal{E} is a direct answer to Q .
EXPLAIN($\mathbf{U}, i, 0, Q$)	INABILITY(\mathbf{R}, j, i, Q)	$Q = Q$.
ATTEMPT($\mathbf{R}, i, t, \mathcal{E}$)	EXPLAIN(\mathbf{U}, j, i, Q')	$Q' = ?\mathbf{W}(F)$ such that $F \subset \mathbf{facts}(\mathcal{E})$.
ATTEMPT($\mathbf{R}, i, t, \mathcal{E}$)	POSITIVE(\mathbf{U}, j, i, A)	$\mathcal{A} \subset \mathcal{E}$ is either a set of facts from the explanation \mathcal{E} or $A = \mathcal{E}$.
ATTEMPT($\mathbf{R}, i, t, \mathcal{E}$)	ASK(\mathbf{U}, j, i, Q')	$\mathbf{facts}(\mathcal{E})$ evokes Q' .
ATTEMPT($\mathbf{R}, i, t, \mathcal{E}$)	NEGATIVE($\mathbf{U}, j, i, \mathcal{E}'$)	$\mathcal{E} = \mathcal{E}'$.
ASK(x, i, t, Q)	ASK(\bar{x}, j, i, Q')	Q implies Q' (cf. Definition 8).
ASK(x, i, t, Q)	ANSWER(\bar{x}, j, i, A)	A is a direct answer to Q .
ANSWER(x, i, t, A)	ASK(\bar{x}, j, i, Q')	A evokes Q' (cf. Definition 7)
ANSWER(\mathbf{R}, i, t, A)	EXPLAIN(\mathbf{U}, j, i, Q')	A evokes Q' (cf. Definition 7)
ASK(x, i, t, Q)	ANSWER(\bar{x}, j, i, \emptyset)	no condition Q .
ANSWER(x, i, t, \emptyset)	no reply	no condition
NEGATIVE($\mathbf{U}, i, t, \mathcal{E}$)	no reply	no condition
POSITIVE(\mathbf{U}, i, t, A)	no reply	no condition
INABILITY(\mathbf{R}, i, t, Q)	no reply	no condition

Table 4: The replies and their semantic conditions. Q is a question, F is a fact, \mathcal{E} is an explanation, $x \in \{\mathbf{U}, \mathbf{R}\}$ and $\bar{x} \in \{\mathbf{U}, \mathbf{R}\} \setminus x$ and $i, t, j \in \mathbb{N}$.

These stores are of crucial importance as summarized hereafter:

- Keep a clear view of the **User**'s state of understanding so he/she can backtrack and request more explanations.
- Judge the success of the explanatory dialogue. For instance, imagine that the understanding store contains a fact F , this indicates that the **User** has not understood F . Now, if the **User** utters a POSITIVE($\mathbf{U}, i, t, \mathcal{E}$) then $\mathbf{explanandum}(\mathcal{E}) = F$ will be deleted from the understanding store. If the dialogue terminates at this point then the explanatory dialogue will be judged successful as we will see in Subsection 4.5.
- Track the consistency of the explanations. For example, imagine that

the **Reasoner** advances the explanation \mathcal{E} and he is in the same time committed to a fact F' which is in contradiction with \mathcal{E} , this would be contradictory.

- Avoid circular explanations. This means that it is forbidden to explain F by an explanation that contains a fact F' which is not yet understood.

Let us formally introduce the notion of stores.

Definition 16 (Stores). *Given a dialogue D_n with a background knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$. A store \mathcal{S} is a set of facts $\mathcal{S} \subseteq \mathcal{F}$. We denote by \mathcal{S}_x^y such that $x \in \{\mathbf{U}, \mathbf{R}\}$ and $y \in \{\mathbf{C}, \mathbf{D}\}$ the store of type y for the participant x at an arbitrary instant within the dialogue D_n . If $y = \mathbf{C}$ then $\mathcal{S}_x^{\mathbf{C}}$ is a commitment store, otherwise it is an understanding store. Note that when $y = \mathbf{D}$ then necessarily $x = \mathbf{U}$ (the **Reasoner** has no understanding store).*

Since a store is a set of facts then the notion of inconsistency defined in Section 2 applies here. Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, a store \mathcal{S}_x^y is inconsistent if and only if $\mathbf{Cl}_{\mathcal{R}}(\mathcal{S}_x^y) \models \perp$.

A store is altered by advancing utterances within the dialogue. For instance, when the **Reasoner** answers a question then the **Reasoner** is committed to the truthfulness of what it said. In what follows we present the effect rules for each store.

Definition 17 (Effect rules). *Let D_n be a dialogue, the stores are altered as follows:*

- If $u_n = \text{EXPLAIN}(\mathbf{U}, i, t, Q = ? \mathbf{W}(F))$ then:
 - $\mathcal{S}_U^{\mathbf{D}} = \mathcal{S}_U^{\mathbf{D}} \cup \{F\}$.
 - $\mathcal{S}_U^{\mathbf{C}} = \mathcal{S}_U^{\mathbf{C}} \cup \{F\} \cup \text{pres}(Q)$.
 - $\mathcal{S}_R^{\mathbf{C}} = \mathcal{S}_R^{\mathbf{C}} \cup \{F\} \cup \text{pres}(Q)$.
- If $u_n = \text{ATTEMPT}(\mathbf{R}, i, t, \mathcal{E})$ then $\mathcal{S}_R^{\mathbf{C}} = \mathcal{S}_R^{\mathbf{C}} \cup \text{facts}(\mathcal{E})$.
- If $u_n = \text{ASK}(x, i, t, Q)$ then $\mathcal{S}_x^{\mathbf{C}} = \mathcal{S}_x^{\mathbf{C}} \cup \text{pres}(Q)$.

- If $u_n = \text{ANSWER}(x, i, t, A)$ then $\mathcal{S}_x^c = \mathcal{S}_x^c \cup \{A\} \cup \text{pres}(Q)$ such that $Q = \text{content}(u_t)$ ⁷.
- If $u_n = \text{NEGATIVE}(\mathbf{U}, i, t, \mathcal{E})$ then $\mathcal{S}_\mathbf{U}^d = \mathcal{S}_\mathbf{U}^d$.
- If $u_n = \text{POSITIVE}(\mathbf{U}, i, t, \mathcal{E})$ then $\mathcal{S}_\mathbf{U}^c = \mathcal{S}_\mathbf{U}^c \cup \{\mathcal{E}\}$, $\mathcal{S}_\mathbf{U}^d = \mathcal{S}_\mathbf{U}^d \setminus \{\text{explanandum}(\mathcal{E})\}$.

Recall that $\text{pres}(Q)$ is the presuppositions of the questions Q , $\text{facts}(\mathcal{E})$ is the factual part of the explanation \mathcal{E} and $\text{explanandum}(\mathcal{E})$ is the object of the explanation \mathcal{E} (cf. Section 3).

When the **User** asks a question then he is committed to the presupposition of the question implicitly. When the **User** or the **Reasoner** answers then they are committed to the answer and to the presupposition of the question they are answering to. When the **User** asks for an explanation $\text{EXPLAIN}(\mathbf{U}, i, j, Q = ?\mathbf{W}(F))$ we add A to $\mathcal{S}_\mathbf{U}^d$ because the two participants know that F is true and one of them is asking for an account about it. When the **User** acknowledges understanding of \mathcal{E} which is an explanation then we revoke the explanandum of the explanation from $\mathcal{S}_\mathbf{U}^d$. This means that the **User** declares that he could understand the explanandum thanks to the explanation. At his point, he/she becomes committed to the truthfulness of the explanation. Therefore we add it to the commitment store. Note that he/she are always committed to the explanandum itself from the beginning. If he dis-acknowledges understanding then we do nothing.

As mentioned these stores will be used to maintain the coherence of the dialogue, hence they will be used in regulating the advancement of utterance in the dialogue alongside with to the semantic conditions provided in the previous section. The following rules complement the rules of syntactical validity (cf. Definition 15).

Definition 18 (Semantic validity). *Let D_n be a syntactically valid dialogue. The dialogue D_n is semantically valid if and only if:*

⁷ $\text{content}(u_t)$ is the question posed in u_t .

(R_4) all utterances $u \in D_n$ are semantically valid (cf. Table 3).

(R_5) every reply u to u' in D_n is semantically valid (cf. Table 4).

(R_6) for any u_i in D_n such that $\mathbf{target}(u_i) = \mathbf{target}(u_n)$ and $\mathbf{loc}(u_i) = \mathbf{POSITIVE}$ then $\mathbf{loc}(u_n) \neq \mathbf{NEGATIVE}$.

(R_7) all commitment stores are consistent.

Note that the empty dialogue D_0 is semantically valid.

The rules R_4 and R_5 insure semantic validity of utterances and replies. R_6 dictates that the **Reasoner** should not give an explanation in which some parts are not understood by the **User**. Finally, R_7 uses the commitment stores to constraint utterance’s validity within the dialogue. It dictates that D_n would not be legal if the utterance u_n introduces inconsistencies in the commitment stores. Note that this rule does not imposes consistency of understanding store because it is possible that one may not be able to understand different things which can be inconsistent.

4.4. Adding Argumentation

In the previous section we assumed that the background knowledge over which the dialogue is carried on is *consistent*. This is not always the case since the knowledge base of the **Reasoner** can hold inconsistent knowledge due to different causes. Concerning our practical setting, this knowledge base is collectively built by several knowledge engineers from different sites of the project. Due to various causes (errors in the factual information due to typos, erroneous databases/excel files, incomplete facts, unspoken obvious information “everybody knows” etc.) the collectively built knowledge base is prone to *inconsistencies*.

Logic-based argumentation is well-used formalism to reason under inconsistency (Martinez et al., 2014; Croitoru & Vesic, 2013). Unlike classical logic it allows to derive queries from a knowledge base even if it contains contradictions. This makes it a suitable tool to handle inconsistency in our case.

In this section we extend the EDS explanatory dialogue system with argumentative faculties, we aim at defining a minimal extension that includes argumentative locutions while keeping the dialogue model as simple as possible. We consider a subset of argumentative locutions from Prakken (2006), which are ARGUE and CONCEDE. These locutions define new utterances which are submitted to the same syntactical validity of Subsection 4.1. To give the semantic of these utterances within an explanatory dialogue, we need to introduce logic-based argumentation. Note that these semantic conditions complement the ones in Subsection 4.2.

Let us define the extended explanatory dialogue system EDS^+ .

Definition 19 (EDS^+). *Let $\mathcal{D}_{sys} = (\mathcal{Pr}, \mathcal{C} = \mathcal{C}_R \cup \mathcal{C}_U, \mathbb{R}, \mathcal{L}^+, \mathcal{K})$ be the explanatory dialogue system EDS. Let $\mathcal{C}' = \mathcal{C} \cup \{\text{ARGUE}, \text{CONCEDE}\}$ and $\mathbb{R}' = \mathbb{R} \cup \{(\text{ARGUE}, \text{ARGUE}), (\text{ARGUE}, \text{CONCEDE}), (\text{ATTEMPT}, \text{ARGUE})\}$. We denote by EDS^+ the extended explanatory dialogue system $\mathcal{D}_{sys} = (\mathcal{Pr}, \mathcal{C}', \mathbb{R}', \mathcal{L}^+, \mathcal{K})$.*

Please note that the new locutions are available to the **Reasoner** as well as to the **User** ($\{\text{ARGUE}, \text{CONCEDE}\} \subseteq \mathcal{C}_U \cap \mathcal{C}_R$). EDS^+ defines a minimal extension of EDS with argumentative locutions. These argumentative locutions allow the **User** to challenge the explanations advanced by the **Reasoner** (the reply ARGUE to ATTEMPT). They also allow the **Reasoner** to attack the arguments used by the **User** to attack the **Reasoner's** explanation (ARGUE to ARGUE). The locution CONCEDE allows both of the participants to settle the argumentative exchange.

In Prakken (2006), the argumentative dialogue incorporates the aforementioned locutions alongside with CHALLENGE⁸ and RETRACT. The reason to not introduce these locutions to EDS^+ is that because of the authoritative nature of our dialogue. In fact, the **User** is often an expert that asks for explanation in order to prob the content of the knowledge base. This expert has the authority in his/her domain, thus a locution CHALLENGE issued by **Reasoner** would not

⁸In fact, this locution is denoted as WHY in Prakken (2006). We prefer to use CHALLENGE in order to avoid confusion between the WHY that challenges and the WHY that asks for an explanation.

be adequate. In addition, it would have the same effect as ARGUE. The locution RETRACT follows the CHALLENGE locution, thus there is no need for RETRACT if CHALLENGE is omitted.

The utterances for ARGUE and CONCEDE are defined as it is indicated in Table 6 column “utterances”. These utterances follow the usual syntactical validity as defined in Definitions 10, 11 and 14.

To update the semantic conditions, we first need to introduce the notion of *argument* and *attack* between arguments. We follow Croitoru & Vesic (2013) in defining argumentation over *Datalog* \pm by instantiating (Dung, 1995).

Definition 20 (Argumentation framework). *Given an inconsistent background knowledge $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$. The corresponding argumentation framework $\mathcal{AF}_{\mathcal{K}}$ of \mathcal{K} is a tuple $(\mathbf{Arg}, \mathbf{Att})$ where \mathbf{Arg} is the set of arguments that is constructed from \mathcal{F} and \mathbf{Att} is a an asymmetric binary relation over \mathbf{Arg} called attack.*

Argument: *an argument is a sequence $\langle F_0, F_1, \dots, F_n, C \rangle$ such that $F_i, i \in [0, n]$ are consistent sets of facts where $F_0 \in \mathcal{F}$, $\mathbf{Cl}_{\mathcal{R}}(F_{i-1}) \models F_i$, and C is a ground conjunct such that $F_n \models C$. The support (resp. conclusion) of an argument a is denoted as $\mathbf{supp}(a) = F_0$ (resp. $\mathbf{conc}(a) = C$).*

Attack: *an argument a attacks b if and only if $\exists h \in \mathbf{supp}(b)$ such that:*

$$\mathbf{Cl}_{\mathcal{R}}(\{\mathbf{conc}(a), h\}) \models \perp.$$

An argument is a logical derivation from the knowledge base of a given conclusion C . An attack is defined as an inconsistency between the conclusion and the support of two arguments.

Example 17 (Arguments and attack). *Take the following example, “the Durum Wheat is attacked by Fusarium because we observe the toxin Deoxynivalenol”, the argument is: $a = (\{isContaminatedBy(d_1, tox), Deoxynivalenol(tox)\}, isAttackedBy(d_1, Fusarium))$. The argument $b = (\{Examination(d_1, e_1), Negative(e_1)\}, Clean(d_1))$ attacks the argument a . The argument b dictates that d_1 is clean because the examination says so while a assumes that it is contaminated, i.e.*

$\{Clean(d_1), isContaminatedBy(d_1, tox)\}$ is inconsistent.

Since the **User** can use arguments to object on some explanations, let us define when an argument attacks an explanation. Given an explanation \mathcal{E} , an argument a attacks \mathcal{E} if and only if $\exists h \in \mathbf{facts}(\mathcal{E})$ such that $\mathbf{Cl}_{\mathcal{R}}(\{\mathbf{conc}(a), h\}) \models \perp$. Note that $(a, \mathcal{E}) \in \mathbf{Att}$ because \mathcal{E} is not an argument. Note also that an explanation cannot attack an argument.

In Table 5 and 6 we present the new semantic conditions. Note that these semantic conditions do not alter the previous ones, they only extend them. The \mathbf{EDS}^+ only extends the locutions, the reply relation and gives their semantics conditions.

In Table 5: “Utterances”, the utterances and their syntax are presented. Table 5: “Meaning” gives the meaning of the utterance. “Effect” extends the effect rules defined in Definition 17 for each new utterance.

In Table 6: “Replies”, the possible replies for each utterance is shown. Table 6: “Conditions” presents the semantic conditions for each reply and utterance.

Utterances	Meaning	Effects
$\mathbf{ARGUE}(x, i, t, a)$	x attacks the an argument or an explanation	$\mathcal{S}_x^c = \mathcal{S}_x^c \cup \mathbf{supp}(a) \cup \mathbf{conc}(a)$ x becomes committed to the support and the conclusion of a .
$\mathbf{CONCEDE}(x, i, t, a)$	x concedes to the argument a	

Table 5: The new utterances and their meaning alongside with their effects on the stores. $x \in \{\mathbf{U}, \mathbf{R}\}$ and \mathcal{S}_x^c is the commitment store of x .

Utterances	Replies	Conditions
$\mathbf{ATTEMPT}(\mathbf{R}, i, j, \mathcal{E})$	$\mathbf{ARGUE}(\mathbf{U}, j, i, a)$	$a \in \mathbf{Arg}$ and a attacks \mathcal{E}
$\mathbf{ARGUE}(x, i, t, a)$	$\mathbf{ARGUE}(\bar{x}, j, i, b)$	$a, b \in \mathbf{Arg}$ and $(b, a) \in \mathbf{Att}$
$\mathbf{ARGUE}(x, i, t, a)$	$\mathbf{CONCEDE}(\bar{x}, j, i, a)$	$a \in \mathbf{Arg}$

Table 6: New replies and their semantic conditions. $x \in \{\mathbf{U}, \mathbf{R}\}$ and $\bar{x} \in \{\mathbf{U}, \mathbf{R}\} \setminus x$. $i, j \in \mathbf{N}$. $\mathcal{AF}_{\mathcal{K}} = (\mathbf{Att}, \mathbf{Arg})$ is the corresponding argumentation framework of \mathcal{K} .

4.5. Termination and Success

In almost any dialogue system, the termination condition should be specified. In our case the termination condition is user-dependent since our dialogue is a human-machine dialogue. Therefore, the termination should be taken care of in the phase of designing the system.

Since the main purpose of explanatory dialogue is to get the **User** to understand query entailment, an important outcome of the dialogue is the success of explanation. In other words, did the **Reasoner** get the **User** to understand the entailment of the query?

In argumentation dialogues (Prakken, 2006) the outcome is defined with respect to the winner of the dispute, in negotiation dialogue it is defined with respect to the allocated resources at the end. Walton (2011) has proposed the *success criterion* as an outcome of an explanatory dialogue. We follow this idea and we define a successful explanatory dialogue as a dialogue where the **User** understands the entailment of the query. This can be translated formally as follows:

Definition 21 (Success). *Let D_n be a terminated dialogue. If $S_U^D = \emptyset$ then D_n is successful, otherwise it is not successful.*

A successful explanatory dialogue is a dialogue where the **User**'s understanding store is empty. Certainly, we cannot be sure whether the understanding has really taken place but it is one way to quantify the success and failure of an explanatory dialogue.

4.6. Dialogue Example

In this subsection we explain how the formal model applies on Example 1. In Table 7 we follow step by step the advanced utterances and the evolution of the stores.

The columns i and x refers to the stage of the dialogue and the speaker respectively. Note that the stages (14') and (15') are alternatives of the stage (14), they present the second scenario when argumentation intervenes. So the

state of the stores OLD at (14') refers to the state of the stores before advancing the utterance at stage (14).

While the column *Text* refers to the textual utterances, the column *Utterance* presents its formal counterpart. Note that we do not detail the content or the presuppositions of questions for each utterance for simplicity reasons. We give just an example to illustrate the idea. For instance, $F_1 = \text{performs}(\text{stubling}, S_1)$ and the presuppositions of the question Q_1 is “the stubble breaking is a process and S_1 is a soil”. Formally, $\text{pres}(Q_1) = \{\text{process}(\text{stubling}), \text{sol}(S_1)\}$. The rest of the column presents the commitment store of R, commitment store of U and the understanding store of U respectively.

As one may notice, each utterance does not necessarily reply to the immediate precedent, but it can go further and reply to earlier utterances. For instance, the utterance $\text{NEGATIVE}(x, i, 2, \mathcal{E}_1)$ replies to the explanation at stage (2). This dialogue, in fact, corresponds to the class of *liberal dialogues* (as defined by Prakken (2006)) which gives a high flexibility to the participants in exchanging utterances. However, this comes with a computational cost, that is why in the implementation we restrict this flexibility to gain more computational efficiency.

The concept of evocation is present in the dialogue where explanations evoke questions. The question $\text{ASK}(x, i, 4, Q_3 = ?\mathbf{O}(F_3(x)))$ at stage (5) has been evoked by the explanation \mathcal{E}_2 in $\text{ATTEMPT}(x, i, 3, \mathcal{E}_2)$ at stage (4). This is not the case for $\text{ASK}(x, i, 0, Q_4 = ?\mathbf{O}(F_4(x)))$ because its target is 0. A question can be asked even if it is not evoked if and only if one of its presuppositions has been established.

The stores evolve due to utterances advancement following the effect rules of Definition 17. For instance, in stage (1) we added F_1 to all the stores. In stage (14) we revoked F_1 from the understanding store because the user has understood the explanation at stage (13) which explains F_1 , the same thing happens for F_2 . This means that U now understands F_1 and F_2 . At this stage the dialogue ends and it is judged to be successful because the understanding store becomes empty (cf. Definition 21). An alternative course of action is the stages (14') and (15), where U advances an argument against the explanation

\mathcal{E}_4 to which **R** has conceded. At this stage the dialogue ends and it is judged unsuccessful because the understanding store is not empty.

5. Implementation of DALEK Framework

In this section we present the general guidelines that we followed to the explanatory dialogue system. The DALEK framework implements the EDS as well as the EDS^+ . Thus the **User** can start a dialogue of EDS where argumentation is not included or a dialogue of system EDS^+ . Besides this local generality, DALEK is programmed in a way that it captures a standalone argumentation dialogue like the one of Prakken (2006) and stand alone explanatory dialogues of Walton (2011); Arioua & Croitoru (2015). DALEK engages a **User** (or an **Expert**) and the **Reasoner** in a dialogue about the entailment of any query in potentially inconsistent *Datalog* \pm knowledge bases.

The layered architecture of Figure 1 is detailed as follows:

- Level 3 (high level): the graphical user interface.
- Level 2: dialogue manager, configuration structure and stores.
- Level 1: dialogue planner and semantics structure.
- Level 0 (low level): logical model.

As depicted in Figure 1 when the **User** interacts with the *GUI*, the latter communicates with *the dialogue manager* which possesses the *configuration structure* and the *stores*. Then, the dialogue manager, at its turn, communicates with *the semantics structure* through the sub-module “Syntax and semantics handler” and with *the dialogue planner* through the sub-module “Utterance dispatcher”. Next, *the dialogue planner* and *the semantics structure* communicate directly with *the logical model* that uses the *Datalog* \pm GRAAL library (Baget et al., 2015) to query *the knowledge base*.

In what follows we detail each module.

i	x	Text	Utterance	\mathcal{S}_k^c	\mathcal{S}_k^c	\mathcal{S}_k^o
1	U	Why do we perform stubble breaking?	EXPLAIN($x, i, 0, Q_1 = ?\mathbf{W}(F_1)$)	$F_1, \text{pres}(Q_1)$	$F_1, \text{pres}(Q_1)$	F_1
2	R	Stubble breaking is necessary to prepare the soil.	ATTEMPT($x, i, 1, \mathcal{E}_1$)	OLD, facts(\mathcal{E}_1)	OLD	OLD
3	U	Why do we prepare the soil?	EXPLAIN($x, i, 0, Q_2 = ?\mathbf{W}(F_2)$)	OLD, $F_2, \text{pres}(Q_2)$	OLD, $F_2, \text{pres}(Q_2)$	OLD, F_2
4	R	Because there was a cultural precedent on the soil.	ATTEMPT($x, i, 3, \mathcal{E}_2$)	OLD, facts(\mathcal{E}_2)	OLD	OLD
5	U	Which cultural precedent?	ASK($x, i, 4, Q_3 = ?\mathbf{O}(F_3(x))$)	OLD, pres(Q_3)	OLD, pres(Q_3)	OLD
6	R	Sunflower.	ANSWER($x, i, 5, A_1$)	OLD, A_1	OLD, A_1	OLD
7	U	I still don't understand why do we need to prepare the soil.	NEGATIVE($x, i, 4, \mathcal{E}_2$)	OLD	OLD	OLD
8	R	The soil needs to be soften.	ATTEMPT($x, i, 3, \mathcal{E}_3$)	OLD, facts(\mathcal{E}_3)	OLD	OLD
9	U	Okey, I understand.	POSITIVE($x, i, 8, \mathcal{E}_3$)	OLD	OLD, facts(\mathcal{E}_3)	OLD $\setminus F_2$
10	U	But I still don't understand why do we perform stubble breaking.	NEGATIVE($x, i, 2, \mathcal{E}_1$)	OLD	OLD	OLD
11	R	Can you tell me what are the hazards against Durum Wheat?	ASK($x, i, 0, Q_4 = ?\mathbf{O}(F_4(x))$)	OLD, pres(Q_4)	OLD, pres(Q_4)	OLD
12	U	Fungi, viruses, weeds.	ANSWER($x, i, 11, A_2$)	OLD, A_2	OLD, A_2	OLD
13	R	Exactly, stubble breaking is used against weeds. By softening the soil, the seeds of the weeds will germinate and grow. Consequently, they will be easy to eliminate in the next seeding plan.	ATTEMPT($x, i, 1, \mathcal{E}_4$)	OLD, facts(\mathcal{E}_4)	OLD	OLD
14	U	Ok, I understand now why do we perform stubble breaking.	POSITIVE($x, i, 13, \mathcal{E}_4$)	OLD	OLD, facts(\mathcal{E}_4)	OLD $\setminus F_1$
14'	U	This will postpone seeding the Durum Wheat, which is against of what we want to do on this soil (early seeding).	ARGUE($x, i, 13, a$)	OLD, conc(a), supp(a)	OLD	OLD
15'	R	I concede.	CONCEDE($x, i, 15, a$)	OLD	OLD, conc(a), supp(a)	OLD

Table 7: Dialogue example. At stage i we show the content of the stores after advancing the utterance at this stage. The utterances 14' and 15' represents the alternative course of action of Example 1. F_1, F_2 are facts, $F_3(x), F_4(x)$ are conjunct with x as a free variable, $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ are explanations where facts(E_i) is the factual part of \mathcal{E}_i . Q_1, Q_2 are why questions and Q_3, Q_4 are open questions. a is an argument where supp(a) (resp. conc(a)) is its support (resp. conclusion).

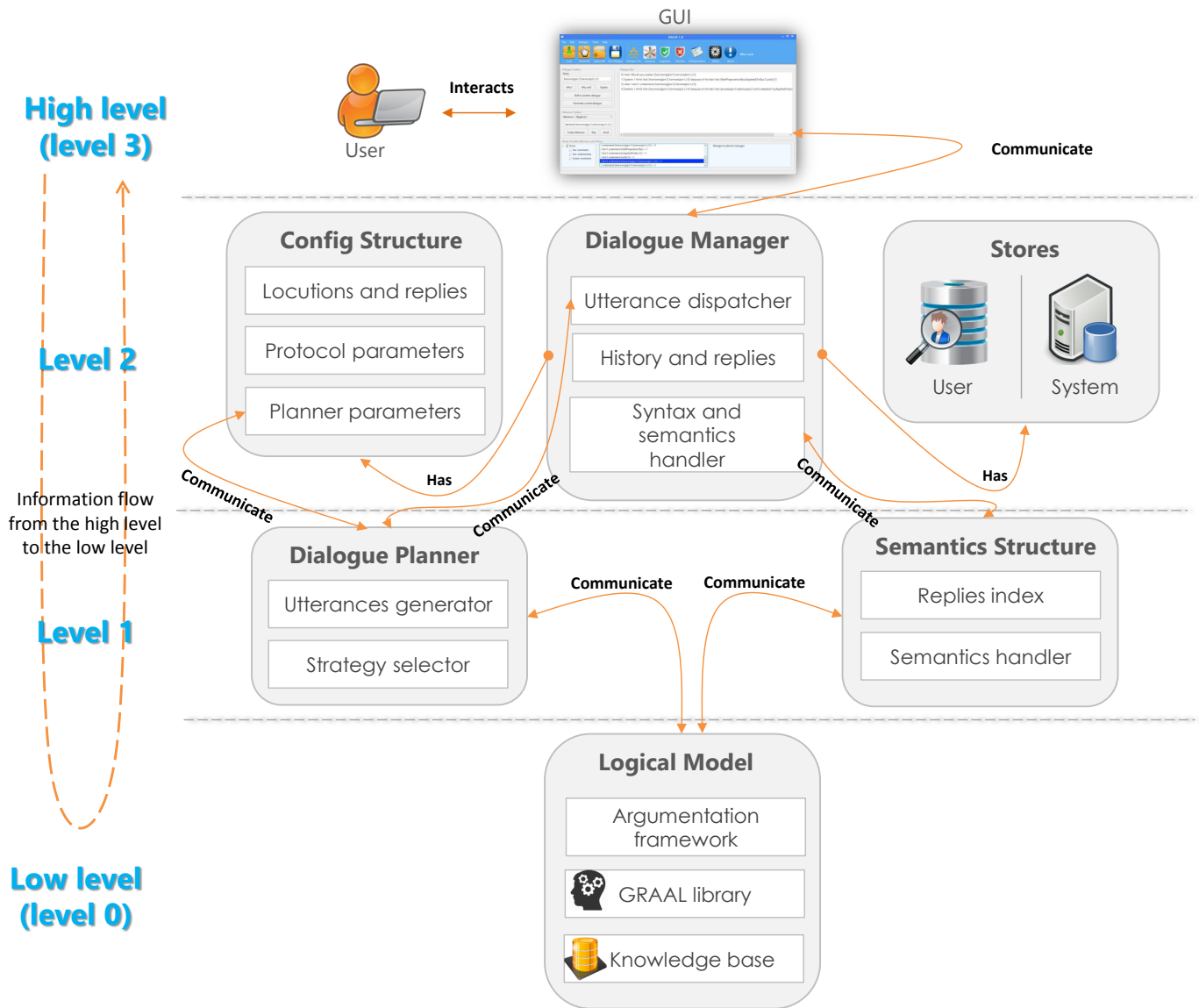


Figure 1: Each layer is composed of modules and each module is composed of sub-modules. The information flow passes from the high level to the low level through the intermediate levels using the “communicate” link between modules. The “has” link symbolizes possession. Note that sub-modules intercommunicate by default.

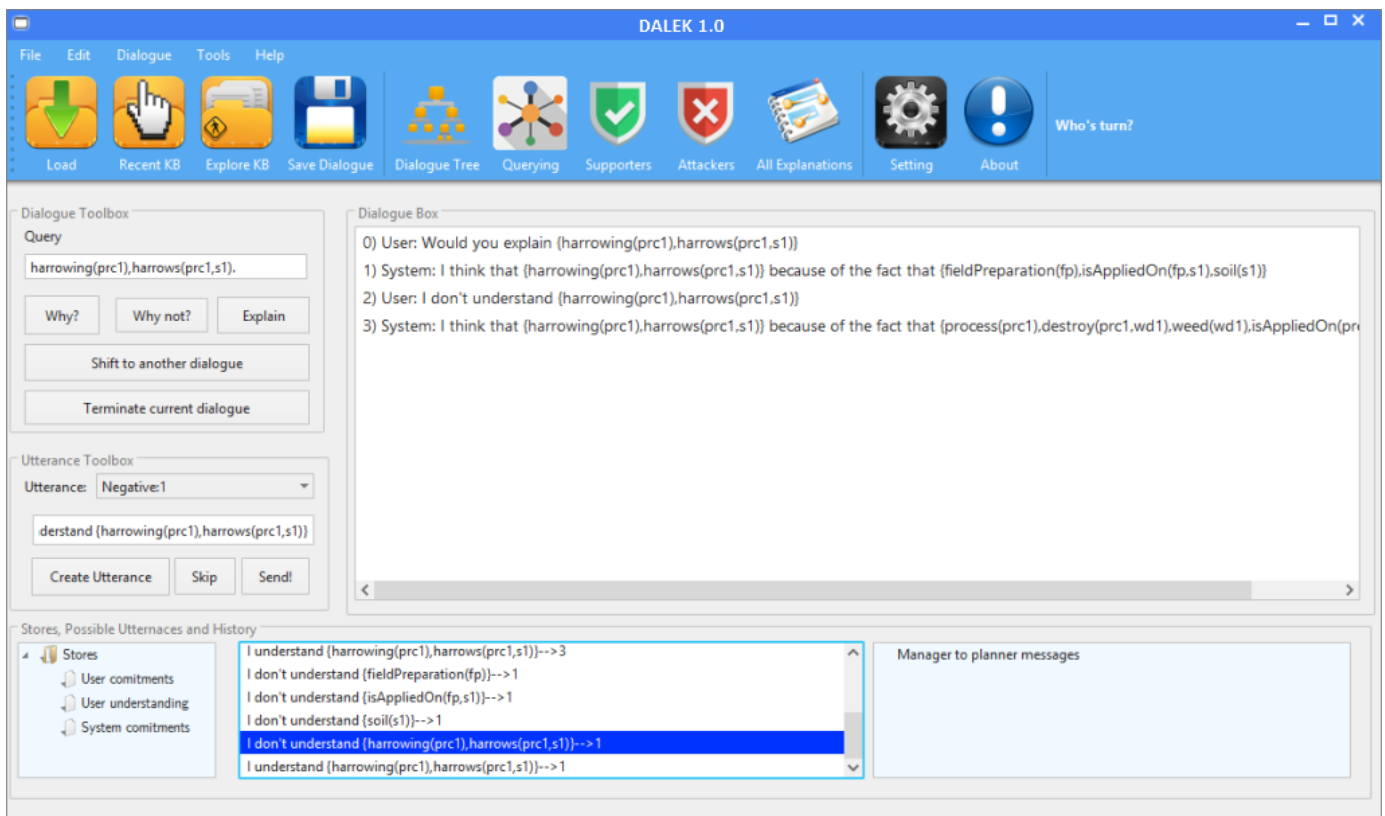


Figure 2: The GUI of DALEK while carrying out a dialogue with a user.

5.1. Configuration structure

This module is responsible for holding the information about the different parameters of the dialogue. It specifies: (1) the set of allowed locutions (e.g. ATTEMPT, POSITIVE, etc.) alongside with their legal replies, (2) the parameters of the protocol, e.g. unique-move, multiple-move, unique-reply, multiple-reply, the participants, etc. and (3) the parameters of the planner, e.g. types of strategies, utterance selection criteria, etc. To facilitate interaction with the **User**, the current version of DALEK adapts a *unique-move and unique-reply protocol*. These settings can be changed in the configuration structure with an additional minor modifications in the planner.

5.2. Stores

As defined in Subsection 4.3 a commitment store is a set of formulae to which a participant is committed to their truthfulness. An understanding store is a set of formulae which a participant has not yet understood. The stores are modified by certain utterances. This module is responsible to manage these stores.

5.3. Dialogue manager

The dialogue manager is the referee between the **User** and the **Reasoner** (i.e. dialogue planner), it dispatches their utterances through the sub-module “Utterance dispatcher” after ensuring their legality. To verify the legality, the dialogue manager communicates with the module semantics structure through the sub-module “Syntax and semantics handler” that makes use of the stores. Here is a brief description of the verification steps⁹.

Syntactical verification. It ensures the legality of any advanced utterance with respect to : (1) legality of the utterance itself, and (2) legality of the reply

⁹For space reasons, each description is not necessarily exhaustive. However, it follows the formalization of Section 4 in defining legal utterances, legal replies, syntactical validity of dialogues, etc.

within the dialogue. The first one checks whether it is the turn of the speaker or not and whether the id of the utterance is correct. It also checks whether the used locution is correct. The second one checks whether the utterance is a correct reply to the previous one by checking membership in the “Locutions and replies” of the configuration structure module.

Semantics verification. It ensures the legality of the utterances with respect to the content. It checks whether the advanced utterance holds a legal content (e.g. EXPLAIN should hold a why question, ATTEMPT should hold an explanation, etc.) and it replies with a legal content. This procedure is ensured by the semantics structure. A final verification is to check whether the commitment stores (resp. understanding store) of the speaker will maintain consistency (resp. avoid circular explanations) after playing the utterance (this is ensured by “Syntax and semantics handler”).

5.4. *Semantics structure*

This structure implements an *operational semantics* of the dialogue. It associates with each reply a procedure that should be called by the dialogue manager to check the legality of the reply. For instance, when presented with an utterance ARGUE(U, 7, 4, b) that responds to ARGUE(R, 4, 3, a), the semantics structure first gets the corresponding procedure (i.e. *negativeToattempt* reply procedure) then checks whether b is an argument, next it verifies whether b attacks a by communicating with the logical model.

5.5. *Dialogue planner (Reasoner)*

This module represents the **Reasoner**. It receives the utterances from the **User** through the dialogue manager and plan the next utterance to advance. The planner in its current state follows a simple profile, a *follow-through* strategy where it tries to answer **User**’s utterances as they come. The planner also performs the following tasks (among others):

Question answering. For each question asked by the **User**, the planner determines the type of question. Then it calls the procedure that is responsible

for answering such question. For instance, open questions are answered by querying the knowledge base using the logical model and then retrieving some tuples that satisfies the query. The why questions are answered by calling the procedure explanation computation in the logical model.

Question generation. Despite the fact that question asking is an important component in EDS⁺. The current status of the planner does not allow for question generation. This functionality would be of great interest when the planner adapts an explanation strategy to facilitate understanding or, in constructing a user model to fit explanations to the **User**. However, the **User** can easily ask questions to the **Reasoner**.

Explanation computation. When an explanation is requested for a query Q , the planer asks the logical model to retrieve the rules and set of facts that can deduce the query in a backward-chaining manner. Using such technique we can insure consistency and minimality of explanations.

5.6. Logical model

Since our explanation dialogue framework uses the *Datalog* \pm language to represent knowledge bases, we have used the library GRAAL (Baget et al., 2015) as a underlying engine for reasoning. The sub-module “Argumentation framework” uses GRAAL and computes argument, counterarguments in the fly using query rewriting techniques.

Inconsistent facts computation. This procedure takes the set of negative constraints and find for each negative constraint the set of facts that triggers it (i.e. a conflict c.f Section 2). If there is such set, it stores it in a file called *conflict*. This file is in fact a physical representation of a hypergraph called the conflict hypergraph (Chomicki & Marcinkowski, 2005).

Argument computation. This task computes an argument given a claim C . It computes by backward-chaining the set of facts involved in the deduction of C . This derivation forms an argument for C .

Counterargument computation. This task receives an argument a and computes its counterargument. It proceeds by getting all the elements of its

hypothesis, then it looks into the graph of conflicts of the knowledge base and try to see whether one of the elements of the hypothesis is involved in any conflict, if so, the procedure constructs a counterargument from the conflict's graph, otherwise it returns an empty set.

6. Pilot Evaluation

The goal of this pilot evaluation is to investigate the gain in terms of new knowledge in presence/absence of explanation dialogue. We have carried out the experiment with two agronomy experts (of similar competences) within the project Dur-Dur. While the number of experts is low in general for a given topic, it is even more difficult to perform studies with domain experts of similar expertise in a project where everybody is chosen to complement the other.

Since DALEK has not yet a natural language processing facility, we have followed the protocol of the *Wizard of Oz* (Kelley, 1984). Of course this part can be replaced in the future by a Web Service implementing the two way translation between Controlled English (Sowa, 2004) and *Datalog* \pm . This pilot evaluation has been carried on a real world knowledge base called the Durum Wheat knowledge base¹⁰. The vocabulary of the knowledge base contains 300 rules and 25 constraints and 900 facts. This knowledge base has been manually authored, thus inconsistencies are presented, in fact we identified 52 conflicts in this knowledge base.

Conditions: The experiment has been conducted with two experts separately. We made sure that the experts do not report each others answers or discussions. The two experts undertook the same experiment.

Description: The expert is presented with a set \mathcal{Q} of 18 queries which are entailed by the knowledge base. The experiment underwent two phases:

- **(Phase 1) without dialogue:** Given \mathcal{Q} the set of 18 queries, in this phase the **Reasoner** asks the expert to indicate for every $Q \in \mathcal{Q}$ whether he agrees

¹⁰Available online at <http://www.lirmm.fr/~arioua/dkb/> where the reader can find a full description of the KB with illustrative examples.

with Q or does not agree *by advancing an argument* supporting or attacking it. To sum up, each query $Q \in \mathcal{Q}$ is associated with an argument a for/against Q .

- **(Phase 2) with dialogue:** In this phase, we allow to further dialogue between the reasoner and the expert as a follow up of Phase 1. The **Reasoner** and the expert are engaged in a dialogue as follows. In Phase 1 for all $Q \in \mathcal{Q}$ the **Reasoner** asks the expert to indicate whether he agrees with Q or not by advancing an argument. Next, in Phase 2, the **Reasoner** answers back to this argument, this gives the possibility to the expert to, again, respond to the recent answer of the **Reasoner**. We consider an explanation dialogue of three stages¹¹. Hence, for every $Q \in \mathcal{Q}$, the query has an associated dialogue d of the form $\langle a_1, e_1, a_2 \rangle$ where e_1 is the explanation advanced by the **Reasoner** in the dialogue and a_1, a_2 are the expert's arguments. In this case Q is associated with more than one argument.

Hypothesis: Our hypothesis is that using a formal model of explanation increases the acquired expert knowledge and removes inconsistencies.

Tables 8 & 9 present the number of rules, facts and negative constraints that have been elicited with Expert 1 & 2 during the two phases. It also presents (last column) the number of inconsistencies (i.e. number of conflicts) in the KB. The KB had *49 inconsistencies* in total before the experiment ¹².

The numbers in the first three columns refer to the new knowledge that has been elicited during phase 2 that we couldn't find in phase 1. In the percentage gain row, we can clearly see that there is 19% new knowledge on average for Expert 1 and 30% for Expert 2 for a dialogue of *only 3 stages*.

In the *inconsistencies* column, we observe that we could remove, independently, 9 with Expert 1 and 12 with Expert 2 in phase 1 (without dialogue), which results in a decrease of inconsistency from **49 to 40** and from **49 to 37** respectively. We observe that the inconsistencies have been reduced signif-

¹¹Note that the reason to limit the dialogue to three stages is to have a uniform and equitable evaluation with respect to all queries.

¹²n/a means that we do not consider the gain.

Expert 1	Rules	Facts	Constraints	Inconsistencies
Phase 1	23	9	8	40
Phase 2	23+6	9+2	8+2	33
% gain	20%	18%	20%	n/a

Table 8: The gain of new knowledge for Expert 1.

icantly in phase 2 (with dialogue) where it goes from **49 to 33** with Expert 1 and from **49 to 28** for Expert 2. The difference in the reduced inconsistencies between phase 1 and phase 2 is due to the significant exchange that has taken place between the system and the experts in phase 2.

To illustrate how inconsistencies are reduced, let us give an example. In the Durum Wheat knowledge base it is stated that: straw cereal precedent and sunflower precedent are used. The problem is that we cannot use the two precedents together. So this is a conflict in the knowledge base. In Table 11 the expert says that the aim is to reduce the use of chemical inputs (herbicide, fertilizers, etc.) and to fight against weed. These information when added to the knowledge base have allowed to conclude that the precedent in question is sunflower. Thus the conflict we had before has been solved, thus the straw cereal precedent is dismissed.

The result of this pilot evaluation presents an encouraging indication of the usefulness of our framework. The evaluation has shown the added value of dialogue usage in knowledge acquisition within our framework. This pilot experiment can serve as basis for numerous extensions. First a belief revision module could be implemented in order to integrate the knowledge automatically without human intervention. Second, in order to run the experiment at a larger scale one could split the knowledge base in relevant modules and have experts of a given expertise work only on their expertise module.

7. Evaluation of the Explanatory Power

The core aspect of our approach is to allow obtaining new knowledge and handling inconsistency through explanation dialogues. This has been demon-

Expert 2	Rules	Facts	Constraints	Inconsistencies
Phase 1	17	11	6	37
Phase 2	17+14	11+9	6+2	28
% gain	45%	45%	33%	n/a

Table 9: The gain of new knowledge for Expert 2.

Queries	Decision	Arguments	Exp
Use Precedent Maize	No	The precedent Maize (corn) is riskier than sunflower for DON (see arvalis grids). It leaves bigger stems than sunflower. From economical perspective, we must see the pedo-climatic context to decide which one to use (Maize or sunflower).	1
Use Precedent Maize	No	Although being at the rotation head, Maize is not a good precedent for Durum Wheat. In fact, Maize is susceptible to Fusarium which survives in crop residues and is transmitted to wheat. This disease results in the production of mycotoxins, toxic molecules for humans.	2

Table 10: An example of experts' responses in phase 1. The decision refers to whether the expert agrees with Q or not.

Queries	Decision	Dialogue
Use a straw cereal precedent	No	<p>1) User: <i>Since the aim is to reduce the use of chemical inputs, we are looking for a precedent that will allow us to fight against weed, provide nitrogen and reduce diseases. Quite the opposite of a cereal that will entertain a specialization of weed Flores and leave little nitrogen in the soil and favor certain disease like septoria.</i></p> <p>2) Reasoner: <i>The introduction of such precedent limits the pressure of selection and generates a bigger range of flora with less density, thus facilitating weed control, hence less cost (financial + reduction of herbicides).</i></p> <p>3) User: <i>I do not really agree. In doing so it promotes a certain specialization of adventitious flora. This flora is going to be increasingly high and increasingly difficult to manage because it will have the same date of exercise of wheat.</i></p>

Table 11: An snippet of an explanation dialogue between **User 1** and the **Reasoner** (phase 2).

strated by the pilot evaluation in the previous section. In the pilot evaluation we have shown how explanation dialogue plays an important role in correcting inconsistencies. In fact, explanation dialogue helps the expert to better under-

stand the content of the knowledge base which provides him/her with a clear view on what might be inconsistent or not. This will eventually enhance inconsistency resolution. Therefore, ensuring that the **Reasoner** provides clear and understandable explanations is very important for what comes after, i.e. correcting and resolving inconsistencies.

In this section, we demonstrate the explanatory power of our model by means of a new proof of concept experimentation carried out with *four experts* who are not experts in Agronomy but in Computer Science (two experts) and Durum Wheat Transformation (two experts). The goal is to show whether or not the generated explanation dialogues would help these experts to understand some basic concepts of Agronomy. It is to be highlighted that the explanation dialogue have been directly carried on the DALEK implementation (the prototype that implements the model). This demonstrates the practical importance of our work.

In this evaluation, each participant is presented with a set of 5 queries that are entailed from the knowledge base under the inconsistency handling semantics. Note that the queries were not generated automatically by the system, DALEK is not designed to do so, but the queries were picked by us for experimentation purposes. These queries are domain-specific and they are presented hereafter:

- (Q_1) Increasing the fertilizer's dose from 40U to 50U on tiller stage does not improve the quality of the durum wheat.
- (Q_2) In the conventional farming system of France it is prohibited to use Corn as precedent.
- (Q_3) In the conventional farming system of France the variety Pescadou is preferred on the variety Miradoux.
- (Q_4) In the conventional farming system of France stubble breaking is a very important step.
- (Q_5) In the conventional farming system of France it is important to make 3 fungicide dozes.

It is to be noted that all the participants are asked the same 5 queries. We recall that the goal of such experiment is to test whether the explanation mechanism of our model is powerful enough to provide clear and understandable explanations.

When the participant is presented with a query, he/she can request an explanation by asking a *why question* or ask other questions, i.e. *whether, who and which*. Then, the **Reasoner** answers to the question as described in the semantics in Section 4.2, page 27. The **Reasoner** also indicates if it has no explanation/answer. Accordingly, the participant is asked to evaluate the explanation advanced by the system on a scale of:

{very clear, clear, so so, not clear, not clear at all}

then, the participant can continue the dialogue according to the protocol of the dialogue model explained in Section 4. For time constraints the dialogues were limited to 10 stages for each query. Each participant undertakes 5 dialogues (dialogue per query). The sessions were done individually with each participant to avoid group influence. We draw the attention of the reader to the following important point. Since DALEK has no natural language processing module currently, the interaction between the participant and DALEK was done with the help of a third-party human operator who translates the natural text advanced by the participant into *Datalog±* and vice versa.

Table 12 presents the statistics of each participant. *D1-D5* denote the dialogues. The numbers in each column represent the number of stages of that dialogue (e.g. 7 stages for P1 in D1). *Nb exp* refers to the total number of explanations advanced by the **Reasoner** which is the sum of the number of explanations in each dialogue. *Nb eval* refers to the total evaluations put by the participant. This column is presented to show that the participants have not missed the evaluation of any explanation. From the statistics, we observe on average that each participant has undertaken 5 dialogues of 5 stages in which he/she receives on average 9 explanations (in total) and he/she evaluates them all. All in all, the **Reasoner** advanced 40 explanations (possibly duplicated) for

Participant	D1	D2	D3	D4	D5	Nb exp	Nb eval
P1	7	7	7	6	7	12	12
P2	3	5	5	3	5	7	7
P3	5	7	3	7	5	11	11
P4	4	4	4	4	4	10	10
Average	4	5	5	5	5	9	9

Table 12: Statistics of each participant.

Participant	Very clear	Clear	So so	Not clear	Not clear at all
P1	1	12	5	1	0
P2	4	1	3	0	1
P3	4	5	1	0	1
P4	5	3	2	0	0

Table 13: Frequency table for evaluations per participant.

the 5 participants. Please note that these statistics are descriptive, therefore other indicators could have been used, however the goal is not inferential but rather exploratory to give the reader a global picture on the proof of concept experiment.

Table 13 presents the statistics of evaluations per participant. We can observe that on average the data is skewed to the left which means that the participants find most of the explanations either *very clear* or *clear*. Some explanations were evaluated as *so so*, however the interesting part is that only 3 explanations out of 40 have been evaluated negatively (*not clear* and *not clear at all*). These observations are confirmed in the frequency table (Table 14) where we show how often the participants find the explanations *very clear*, *clear*, *so so*, *not clear* and *not clear at all* respectively.

The median as one can clearly see is *clear*. However, there are not much data to draw a significantly reliable **general** conclusion. Even though, the preliminary results are encouraging when taken within the context of the ANR DUR-DUR project at least with respect to the participants in that project.

	Very clear	Clear	So so	Not clear	Not clear at all
Nb exp	14	18	5	1	2

Table 14: Frequency table for evaluations. Median is **Clear**.

8. Related Work and Discussion

Our work is in the intersection of different domains in Expert and Intelligent Systems with respect to different aspects. This section is far from being exhaustive on all related state of the art, so in what follows we present each domain by showing the closest work to ours in the literature. We point how our work contributes and differs from the state of the art.

Explanation in KBS. Explanation is central to Knowledge-based System because it reinforces trust and transparency and helps the user in problem solving as argued in Darlington (2013); Metzler & Martincic (1998); Haynes et al. (2009). In the first and second generation of Expert Systems, explanation has three forms: trace explanations, strategic explanations and deep explanations (Moulin et al., 2002). In trace explanations the system provides an explanation by producing an execution trace, a sequence of inferences starting from a set of facts leading to the query. Strategic explanations contextualize the explanation by exhibiting the problem-solving strategy of the system. Deep explanations use a user model to know what the user knows in order to fill in the gaps and provide a custom-tailored explanation. Such explanation approaches¹³ are not robust against inconsistency. There are no means by which the system can explain whether the result is inconsistent or not and why the knowledge base is inconsistent. This happens for two reasons, (1) inconsistency is not accounted for and not dealt with, (2) as a consequence of the first reason, no explanation model has been dedicated for such case. Our approach accounts for inconsistency as described in Section 2 and provide explanations. Furthermore, the explanation provided is dialogical which provides a new insight with respect to

¹³See Moulin et al. (2002) for more details.

the types of explanation cited above.

Explanation dialogues in KBS. Dialogue systems for explanation have been recently studied in Walton (2007, 2008, 2009, 2011, 2016) which lay the foundation of dialectical explanatory dialogues. These works advocate the idea that explanatory dialogues are separate type of dialogues as much as other types of dialogues (e.g. inquiry, argumentation, etc.). The proposed model in Walton (2011, 2016) uses speech act of requesting an explanation, attempting an explanation, informing about intelligibly or unintelligibly of explanations. The model is not fully formalized and presents general guidelines about its instantiation in practical cases. Given our specific context, the model EDS extends Walton’s model on different aspects. It introduces questions as an important component within the dialogue. It also integrates argumentation faculties to handle inconsistent background knowledge bases. This very point is also a departure from Arioua & Croitoru (2015). EDS is fully formalized which may allow further formal investigation. Furthermore, the protocol of EDS allows for nested explanation requests and liberal turn taking where the participants can retrace and continue the dialogue following other directions. Bex & Walton (2011); Bex et al. (2012); Letia & Groza (2012) propose a hybrid model of explanation and argumentation. Their aim is to be able to distinguish explanations and arguments in the context of a dialogue to avoid “the mistake of treating something as fallacious while it is not.” as noted in Bex & Walton (2011), (p. 1). Despite the importance of such endeavor, our context is different from theirs. Firstly, we consider a logical setting where logic-based argumentation is used. Secondly, we use argumentation and explanation to better expose the content of the inconsistent knowledge base. Arguments are used to attack explanations to eventually update the knowledge base with new knowledge. The works of Cawsey (1992); Maybury (1992); Moore & Moore (1995); Minock & Chu (1996) were mainly used in explaining the output of the system for *persuasion purposes* to see the impact of explanation on the acceptance of the KBS by users. Unfortunately, these approaches lack a formal characterization of their components

which makes them difficult to study and investigate. Consequently, the explanatory dialogue seems to be persuasive rather than explanatory. In addition, it is not clear to which extent these approaches can be reused to handle the context of knowledge acquisition within inconsistent knowledge bases.

Inconsistency handling in KBS. To overcome the problem of inconsistency in KBS, inconsistency-tolerant systems have been proposed (Lembo et al., 2015; Bourgaux, 2016; Du & Qi, 2015; Bienvenu & Rosati, 2013; Bienvenu, 2012; Lembo et al., 2010). The basic idea which is common among all these systems is that they locate the inconsistent part of the knowledge base using a set of integrity constraints then they distinguish the consistent part. The system *repairs* the knowledge base by only keeping the consistent part and throwing away the inconsistent part. Different strategies are adapted to perform this process, for instance cardinality-based, intersection-based, just to name a few.¹⁴ The common drawback of these approaches is in the repairing mechanism. These approaches are lossy in the sense that valuable knowledge can be thrown away. The new contribution in our model is that explanation dialogue is used to resolve inconsistency by engaging the expert in the discovery and resolution of inconsistencies. As a result we avoid losing important and valuable pieces of knowledge.

Explanation in presence of inconsistency. There has been an increasing interest in equipping inconsistency-tolerant systems with explanation facilities. The closest approaches are those that are dedicated to Description Logics knowledge bases (DL KBs). In McGuinness & Borgida (1995) the authors addressed the problem of explaining subsumption and non-subsumption in a coherent and satisfiable DL knowledge base using *formal proofs as explanation* while other proposals such as Schlobach et al. (2003); Borgida et al. (2008) have used *Axiom pinpointing* and *Concept pinpointing* as explanation to highlight contradictions within an unsatisfiable and incoherent DL KB. Another proposal is the so-

¹⁴See Baget et al. (2016) for a unified framework and Bertossi et al. (2005) for a full survey.

called *justification-oriented proofs* (Horridge et al., 2010) in which the authors proposed a *proof-like* explanation without the need for a deduction system. These approaches are fundamentally different from ours. Their aim is diagnostic more than explanatory, they explain to the user why the knowledge base is inconsistent. They fail, as opposed to our approach, to provide domain-specific explanations. In addition, our model provides domain-specific explanations in a dialogical manner which is another departure from their approaches. We find the same difference between our approach and explanation approaches like Du et al. (2015); Bienvenu et al. (2015); Arioua et al. (2015); García et al. (2013) which provide explanations as a way to justify query answering under inconsistency. These are all oneshot-based explanations where no interaction with the user is considered. We draw the attention that this aspect has been already investigated in a work of ours in which we introduced the concept of explanation dialogue to explain inconsistency (Arioua et al., 2014). In this work we generalize this approach from different aspects and we show how this general framework can serve as a method to acquire more knowledge to resolve inconsistency when the knowledge to be thrown away is costly.

Sonntag & Theobald (2010) is the closest work to ours. It considers an uncertain RDF knowledge base and provides a dialogue system for querying the knowledge base, the system can generate explanations in form of graphs. The dialogue system is rich when it comes to handling natural language and different input methods, hence multimodal. All of that is provided even if the knowledge base is inconsistent. The contribution of our work with respect to theirs is described as follows: (1) **formal model**: we propose a formal model of *explanation dialogues* that offers a clear dichotomy between syntax and semantics, this makes the system extensible and reusable. Their dialogue model is a *Question-Answering* (QA) model¹⁵ in which the user queries the KB in a QA dialogue. The EDS model is fundamentally different in the sense that our dialogue is an explanation dialogue which is capable of capturing normal

¹⁵Please see Sonntag & Theobald (2010), Section 5 for an example.

QA dialogue. Plus, questions in EDS are formally defined which makes them extensible to incorporate other types of questions. In EDS the system can ask the user questions to acquire knowledge from the expert. The argumentative capacities are absent in their work as apposed to EDS. (2) **expressiveness**: the knowledge representation logical language we use is more expressive then theirs (OWL) as shown in Mugnier & Thomazo (2014). (3) **dichotomy**: the dialogue we propose presents a dichotomy of syntax and semantics to reinforce usability and to offer a natural explanatory exchange. In their model it is not clear what are the rules of the QA dialogue and when and how the user and the system can interact with each others. Unfortunately this would put difficulties in reusing the system as opposed to the EDS model that can be implemented only by respecting the semantics and the syntax of the dialogue.

9. Conclusion and Future Work

The central task that Intelligent and Expert Systems are set to perform is reasoning. Such task becomes harder when inconsistency overtakes on some parts of the knowledge base. As discussed in the previous section, handling inconsistency is well studied in the tradition of Knowledge Representation and Reasoning and starts getting its way into different aspects of Expert Systems. Our work comes within such spirit, in which we reconcile Knowledge Acquisition with Inconsistency Handling. Such reconciliation takes another important dimension into account in Expert Systems, which is *explanation and dialogues*. So given the context of Knowledge Acquisition, Inconsistency Handling and Explanation our work comes to establish *a framework of inconsistency handling through knowledge acquisition by means of explanation dialogues*. The claim behind our proposal is that explanation dialogue can play a central role in modern knowledge-based systems. Firstly, it can be used *solo* where the aim is to reinforce transparency between the system and the user, especially when it comes to inconsistent knowledge bases. Secondly, It can be involved in a triad of *Explanation-Acquisition-Resolution* where not only transparency is reinforced

but the quality of the knowledge is improved by means of expert-guided inconsistency resolution as opposed to reparation techniques which are lossy by nature.

To summarize the contributions of the paper. We have proposed a formal model of explanatory dialogues called **EDS**. The aim of this model is to give a formal account of explanatory dialogues in inconsistent knowledge bases. The **EDS** model aims at facilitating the understanding of query entailment in these knowledge bases and facilitates knowledge acquisitions under such setting. To prove the relevance of the formal model we have implemented a system called **DALEK** that realizes the formal framework and propose other general capacities. To evaluate and validate the **DALEK** framework we have carried out a pilot evaluation with agronomy experts in the context of the ANR Dur-Dur project. The evaluation gives a promising indication of this approach as far as our application setting is concerned. Another proof-of-concept experimentation revealed how other experts find the explanation dialogue usable and understandable.

Despite the framework is meant to be general some limitations have to be addressed. The questioning mechanism is extensible in the sense that more questions can be incorporated by only integrating a proper syntax and semantics. A good starting point would be integrating “how?” based on the work of Jaworski (2009). However, other questions may pose some difficulties, one could think of counterfactuals like “what would be the case if X were to be Y?” or time-related questions like “when?” or even a mixture like “when would be...?”. These questions need a framework more expressive than the one we propose (it incorporates temporal modalities, counterfactual conditionals, etc.). Any future research should take into account the expressiveness and efficiency problem by investigating the decidability of query answering, which is the task that we will undertake to improve the flexibility of our framework before extending the dialogue model.

No matter how much the dialogue model is flexible, being unable to automatically adapt the user’s feedback about inconsistency resolution is still a limitation in the sense that it would involve other knowledge engineers to per-

form such task. To overcome such limitation, we aim for future work to develop a natural language processing module for DALEK by implementing a two-way translation between Controlled English (Sowa, 2004) and *Datalog* \pm . In addition, evolving and revising the ontology with respect to the expert’s input should be taken care of, as a starting point we will make use of the approaches proposed in Ma et al. (2010); Gom (2013). Another promising research direction is the formal study of *explanation strategies* in which the system tries to make the user understand while minimizing/maximizing a payoff function. We will investigate how strategies can be defined in our framework following the literature of game theory and argumentation dialogues. A good starting point is the work on strategic argumentation of Amgoud & Hameurlain (2006); Governatori et al. (2014).

Acknowledgment

The authors acknowledge the support of the French National Research Agency (ANR), grant of the project DURDUR (ANR-13-ALID-0002).

References

- (2013). Ontoarg: A decision support framework for ontology integration based on argumentation. *Expert Systems with Applications*, 40, 1858 – 1870.
- Amgoud, L., & Hameurlain, N. (2006). A formal model for designing dialogue strategies. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)* (pp. 414–416).
- Arioua, A., & Croitoru, M. (2015). Formalizing explanatory dialogues. In *Proceedings of Scalable Uncertainty Management - 9th International Conference, SUM 2015* (pp. 282–297).
- Arioua, A., Tamani, N., & Croitoru, M. (2015). Query answering explanation in inconsistent datalog+/- knowledge bases. In *Proceedings of DEXA’15* (pp. 203–219). Springer.

- Arioua, A., Tamani, N., Croitoru, M., & Buche, P. (2014). Query failure explanation in inconsistent knowledge bases: A dialogical approach. In *Proceedings of SGAI'14* (pp. 119–133). Springer.
- Atkinson, K., Bench-Capon, T., & McBurney, P. (2005). A dialogue game protocol for multi-agent argument over proposals for action. *Autonomous Agents and Multi-Agent Systems*, *11*, 153–171.
- Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the envelope. In *Proceedings of 19th International Joint Conference on Artificial Intelligence, IJCAI 2005*.
- Baget, J., Benferhat, S., Bouraoui, Z., Croitoru, M., Mugnier, M., Papini, O., Rocher, S., & Tabia, K. (2016). A general modifier-based framework for inconsistency-tolerant query answering. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)* (pp. 513–516).
- Baget, J.-F., Leclère, M., Mugnier, M.-L., Rocher, S., & Sipieter, C. (2015). Graal: A toolkit for query answering with existential rules. In *Proceedings of The 9th International Web Rule Symposium, RuleML'15* (pp. 328–344). Springer.
- Baget, J.-F., Leclère, M., Mugnier, M.-L., & Salvat, E. (2011a). On rules with existential variables: Walking the decidability line. *Artificial Intelligence*, *175*, 1620 – 1654.
- Baget, J.-F., Mugnier, M.-L., Rudolph, S., & Thomazo, M. (2011b). Walking the complexity lines for generalized guarded existential rules. In *Proceedings of 22nd International Joint Conference on Artificial Intelligence, IJCAI'11* (pp. 712–717).
- Bertossi, L., Hunter, A., & Schaub, T. (2005). Introduction to inconsistency tolerance. In *Inconsistency Tolerance* (pp. 1–14). Springer.

- Bex, F., Budzyska, K., & Walton, D. (2012). Argumentation and explanation in the context of dialogue. *Explanation-aware Computing ExaCt 2012*, (p. 6).
- Bex, F., & Walton, D. (2011). Combining explanation and argumentation in dialogue. In *the 12th workshop on Computational Models of Natural Argument* (p. 52).
- Bienvenu, M. (2012). On the complexity of consistent query answering in the presence of simple ontologies. In *Proceedings of The 26th AAAI Conference on Artificial Intelligence, AAAI'12*.
- Bienvenu, M., Bourgaux, C., & Goasdoué, F. (2015). Explaining query answers under inconsistency-tolerant semantics over description logic knowledge bases. In *Proceedings of the 28th Description Logics workshop* (pp. 28–39).
- Bienvenu, M., & Rosati, R. (2013). Tractable approximations of consistent query answering for robust ontology-based data access. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013* (pp. 3–9).
- Borgida, A., Calvanese, D., & Rodriguez-Muro, M. (2008). Explanation in the dl-lite family of description logics. In *Proceedings of the 7th international conference on ontologies, databases, and applications of semantics, ODBASE 2008* (pp. 1440–1457). Springer.
- Bourgaux, C. (2016). *Inconsistency Handling in Ontology-Mediated Query Answering*. Ph.D. thesis University of Paris-Saclay, France. URL: <https://tel.archives-ouvertes.fr/tel-01378723>.
- Calì, A., Gottlob, G., & Lukasiewicz, T. (2009). Datalog+/-: a unified approach to ontologies and integrity constraints. In *Proceedings of the 12th International Conference on Database Theory, ICDT'09* (pp. 14–30). ACM.
- Calì, A., Gottlob, G., & Lukasiewicz, T. (2012). 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.

- Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. Autom. Reasoning*, *39*, 385–429.
- Cawsey, A. (1992). *Explanation and interaction: the computer generation of explanatory dialogues*. MIT press.
- Ceri, S., Gottlob, G., & Tanca, L. (1989). What you always wanted to know about datalog (and never dared to ask). *Knowledge and Data Engineering, IEEE Transactions on*, *1*, 146–166.
- Chein, M., & Mugnier, M. (2009). *Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs*. Advanced Information and Knowledge Processing. Springer.
- Chomicki, J., & Marcinkowski, J. (2005). Minimal-change integrity maintenance using tuple deletions. *Information and Computation*, *197*, 90–121.
- Croitoru, M., & Vesic, S. (2013). What can argumentation do for inconsistent ontology query answering? In *Proceedings of SUM'2013* (pp. 15–29). Springer Berlin Heidelberg.
- Darlington, K. (2013). Aspects of intelligent systems explanation. *Universal Journal of Control and Automation*, *1*, 40–51.
- Du, J., & Qi, G. (2015). Tractable computation of representative abox repairs in description logic ontologies. In *Knowledge Science, Engineering and Management* (pp. 28–39). Springer.
- Du, J., Wang, K., & Shen, Y. (2015). Towards tractable and practical abox abduction over inconsistent description logic ontologies. In *Proceedings of AAAI'15* (pp. 1489–1495).
- Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, *77*, 321–357.

- García, A. J., ChesñEvar, C. I., Rotstein, N. D., & Simari, G. R. (2013). Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications*, *40*, 3233–3247.
- Governatori, G., Olivieri, F., Scannapieco, S., Rotolo, A., & Cristani, M. (2014). Strategic argumentation is np-complete. In *Proceedings of the 21st European Conference on Artificial Intelligence, ECAI 2014* (pp. 399–404). IOS Press.
- Gratton, C. (1994). Circular definitions, circular explanations, and infinite regresses. *Argumentation*, *8*, 295–308.
- Hamblin, C. L. (1958). Questions. *Australasian Journal of Philosophy*, *36*, 159–168.
- Haynes, S. R., Cohen, M. A., & Ritter, F. E. (2009). Designs for explaining intelligent agents. *International Journal of Human-Computer Studies*, *67*, 90 – 110.
- Hempel, C. G., & Oppenheim, P. (1948). Studies in the logic of explanation. *Philosophy of science*, (pp. 135–175).
- Hodkinson, I., Wolter, F., & Zakharyashev, M. (2000). Decidable fragments of first-order temporal logics. *Annals of Pure and Applied logic*, *106*, 85–134.
- Horridge, M., Parsia, B., & Sattler, U. (2010). Justification oriented proofs in owl. In *Proceedings of ISWC 2010*. (pp. 354–369). Springer-Verlag.
- Jaworski, W. (2009). The logic of how-questions. *Synthese*, *166*, 133–155.
- Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, *2*, 26–41.
- Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., & Savo, D. F. (2010). Inconsistency-tolerant semantics for description logics. In *Proceedings of RR'10* (pp. 103–117). Springer-Verlag.

- Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., & Savo, D. F. (2015). Inconsistency-tolerant query answering in ontology-based data access. *Web Semantics: Science, Services and Agents on the World Wide Web*, 33, 3–29.
- Letia, I. A., & Groza, A. (2012). Interleaved argumentation and explanation in dialog. In *Proceedings of 12th workshop on Computational Models of Natural Argument* (p. 44).
- Lukasiewicz, T., Martinez, M. V., Pieris, A., & Simari, G. I. (2015). From classical to consistent query answering under existential rules. In *Proceedings of AAAI'15* (pp. 1546–1552).
- Ma, Y., Liu, S., Jin, B., & Xu, G. (2010). Inconsistent ontology revision based on ontology constructs. *Expert Systems with Applications*, 37, 7269 – 7275.
- Mackenzie, J. D. (1979). Question-begging in non-cumulative systems. *Journal of philosophical logic*, 8, 117–133.
- Martinez, M. V., Deagustini, C. A. D., Falappa, M. A., & Simari, G. R. (2014). Inconsistency-tolerant reasoning in datalog+/- ontologies via an argumentative semantics. In *Advances in Artificial Intelligence-IBERAMIA 2014* (pp. 15–27). Springer.
- Maybury, M. T. (1992). Communicative acts for explanation generation. *International Journal of Man-Machine Studies*, 37, 135–172.
- McBurney, P., & Parsons, S. (2009). Dialogue games for agent argumentation. In G. Simari, & I. Rahwan (Eds.), *Argumentation in Artificial Intelligence* (pp. 261–280). Springer US.
- McBurney, P., Parsons, S., & Wooldridge, M. (2002). Desiderata for agent argumentation protocols. In *Proceedings of the 1st international joint conference on Autonomous agents and multiagent systems: part 1* (pp. 402–409). ACM.
- McGuinness, D. L., & Borgida, A. T. (1995). Explaining subsumption in description logics. In *Proceedings of IJCAI'95* (pp. 816–821). Morgan Kaufmann Publishers Inc.

- Metzler, D. P., & Martincic, C. J. (1998). Que: explanation through exploration. *Expert Systems with Applications*, 15, 253 – 263.
- Minock, M. J., & Chu, W. W. (1996). Explanation for cooperative information systems. In *Proceedings of the 9th International Symposium on Methodologies for Intelligent Systems* (pp. 264–273). Springer.
- Moore, J. D., & Moore, J. D. (1995). *Participating in explanatory dialogues: interpreting and responding to questions in context*. MIT press Cambridge, MA.
- Moulin, B., Irandoust, H., Bélanger, M., & Desbordes, G. (2002). Explanation and argumentation capabilities:towards the creation of more persuasive agents. *Artificial Intelligence Review*, 17.
- Mugnier, M. (2011). Ontological query answering with existential rules. In *Proceedings of RR'11* (pp. 2–23).
- Mugnier, M., & Thomazo, M. (2014). An introduction to ontology-based query answering with existential rules. In *Proceedings of RR'14* (pp. 245–278).
- Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2008). Linking data to ontologies. In *Journal on data semantics X*, 133-143. Springer.
- Prakken, H. (2006). Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21, 163–188.
- Preece, A. D. (1993). A new approach to detecting missing knowledge in expert system rule bases. *Int. journal of man-machine studies*, 38, 661–688.
- Schlobach, S., Cornet, R. et al. (2003). Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI* (pp. 355–362). volume 3.

- Sonntag, D., & Theobald, M. (2010). Explanations in dialogue systems through uncertain rdf knowledge bases. In *5th International Workshop on Explanation-Aware Computing, ExaCt'10* (pp. 49–60).
- Sowa, J. F. (2004). *Common logic controlled english*. Technical Report Draft, 24 February 2004, <http://www.jfsowa.com/clce/specs.htm>.
- Walton, D. (2007). Dialogical models of explanation. In *Proceedings of the AAAI Workshop on Explanation-Aware Computing (ExaCt'07)* (pp. 1–9). volume 2007.
- Walton, D. (2008). Can argumentation help ai to understand explanation? *KI*, *22*, 8–11.
- Walton, D. (2009). Explanations and arguments based on practical reasoning. In *The AAAI Workshop on Explanation-Aware Computing (ExaCt'09)* (pp. 72–83).
- Walton, D. (2011). A dialogue system specification for explanation. *Synthese*, *182*, 349–374.
- Walton, D. (2016). A dialogue system for evaluating explanations. In *Argument Evaluation and Evidence* (pp. 69–116). Springer.
- Walton, D., & Krabbe, E. C. (1995). Commitment in dialogue. *State University of New York Press, Albany*, .
- Walton, D. N. (1984). *Logical Dialogue-Games*. University Press of America, Lanham, Maryland.
- Wisniewski, A. (2013). *The Posing of Questions: Logical Foundations of Erotetic Inferences*. Synthese Library. Springer Netherlands.