Properties for a Formal Model of Collaborative Dialogue *

M. Julieta Marcos Marcelo A. Falappa Guillermo R. Simari
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial,
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur,
Avenida Alem 1253,(B8000BCP), Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136
Email: {mjm,mfalappa,grs}@cs.uns.edu.ar

Abstract

We propose a basic set of *desirable properties* for an abstract model of *collaborative dialogue* among agents. The abstraction comprehends the underlying logic of the agents, as well as the interaction protocol. The properties pursue the characterization of finite dialogues, with reasonable conclusions (based on what the participants have said), in which everything said is relevant and everything relevant is said. To this end, two levels of *relevance* (*direct* and *potential*) are defined, based on the notions of *inference* and *abduction*, respectively. Illustrative examples, using mainly the DeLP formalism, are provided.

Keywords: Multi-agent Systems, Collaborative Dialogue, Properties, Relevance, Abduction.

1 INTRODUCTION & BACKGROUND

Agents in a multi-agent system need to communicate for different reasons: to resolve differences of opinion or conflicts of interests, to cooperate for solving problems or finding proofs, or simply to communicate each other about pertinent facts [14]. This gives rise to a variety of dialogue types. The following are some particular types of dialogue, as characterized in [15], which have gained special attention:

Information Seeking Dialogue. One participant seeks the answer to some question(s) from another participant, who is believed to know the answer(s) by the first one.

Inquiry Dialogue. The participants collaborate to answer some question or questions whose answers are not known to any of the participants.

Persuasion Dialogue. One party seeks to persuade another party to adopt a belief or point of view he or she does not currently hold.

Negotiation Dialogue. The participants bargain over the division of some scarce resources in a way acceptable to all, with each individual party aiming to maximize his or her share.

*This work is partially supported by Consejo Nacional de Investigaciones Científicas y Técnicas (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11).

Deliberation Dialogue. Participants collaborate to decide what course of action to take in a certain situation.

A distinction can be made between *collaborative* and *non-collaborative* dialogues. In the former, there is a common goal shared by all the participants: to obtain, from the union of their knowledge, a unified conclusion about certain topic. Thus, collaborative agents (these could be denoted as *unbiased* agents) will expose any knowledge that is relevant for the determination of the conclusion. On the other hand, in a non-collaborative dialogue, the agents may have individual goals which lead them to prefer some conclusions over others. Thus, non-collaborative agents (these could be denoted as *biased* agents) may hide knowledge, being aware of its relevance, because it does not favor the achievement of their individual goals. In the above typology, the first and second are collaborative dialogue types, whereas the third and fourth ones are clearly non-collaborative (the persuader is biased to obtain the predetermined conclusion he is trying to convince the others of, and the negotiating agents are biased to obtain the conclusion that maximize their own profit).

Much work has been done to develop formal models of these dialogues, in order to achieve their automatization. The aim of this work is to formalize some properties that should be satisfied by collaborative-dialogue formal models in general. In order to do that, we will define a *dialogue operator* as a black box that takes the inputs (agents knowledge bases and dialogue topic), and returns the conclusion achieved, together with the sequence of steps that conform the dialogue. It is assumed that a single step of the dialogue consists in an assertion made by one of its participants, and everything that an agent asserts is a subset of its private knowledge base. It is also assumed that there is a common underlying *inference mechanism* which allows to obtain a conclusion about a topic, on the basis of an individual set of knowledge. Based on all these elements, we will characterize the expected behavior of the dialogue operator through its properties. The intuitions behind these properties are simple-minded: we look for finite dialogues with reasonable conclusions (based on what the participants have said), in which everything said is relevant and everything relevant is said. In other words, we point to: a reasonable connection between the steps and the conclusion, the relevance of each step, and the completeness, and finiteness, of the sequence of steps.

Within a context of *incomplete information*, as a dialogue is (since each agent ignores the other agents private knowledge), it may be not so easy to know that certain information is relevant for achieving the conclusion. For example, suppose that agents A_1 and A_2 engage in dialogue to determine whether certain individual x flies or not. A_1 knows that x is a bird and x is green, and A_2 knows that all birds fly and also that all planes fly. Considering the knowledge of both agents as a whole, it is easy to see that they should conclude that x flies, and that the relevant information to achieve that conclusion is "x is a bird" and "all birds fly". However, if we want to define a criterion for an agent to use in any step of the dialogue, in order to determine the relevance of its knowledge, then some questions arise, for instance: how can it be determined, before A_1 says anything, that A_2 has some relevant information (or viceversa)? Is it possible to distinguish, before A_1 says anything, that "all birds fly" is relevant but "all planes fly" is not? In this work, the notion of *abduction* will be used for determining relevance in a context of incomplete information. An abduction mechanism will allow to discover what information (facts) is missing to obtain certain conclusion. Then, a set of knowledge will be considered relevant if its addition changes the current conclusion (*direct relevance*), or if it changes the possible sets of facts that are missing to change the current conclusion (*potential relevance*). As we shall see, this *potential relevance* notion captures the idea of guessing what may be relevant in the future, after some other pieces of information are considered. Reconsidering the situation illustrated above, both "all birds fly" and "all planes fly" will be considered potentially relevant, before A_1 says anything, whereas "x is a bird" will be considered (directly) relevant only after "all birds fly" has been said.

The incompleteness of knowledge, discussed above, is an intrinsic feature of dialogues, caused by the distribution of knowledge among several sources. Another characteristic of dialogues, due to the same cause, is the potential *inconsistency* of knowledge. In this work, we will take into account that different agents may have inconsistent information, although it will be assumed that there is common agreement among the participants on the way this inconsistency is handled (this will be reflected on a unified consolidation function, as well as a common preference criterion in the case of argumentative systems).

This work is organized as follows: in section 2 a brief summary of an argumentative formalism (DeLP), which will be extensively used in examples throughout this paper, is given. In section 3 we define abstract versions of some well known concepts in logic (namely: consistency, consolidation, inference and abduction) so that we can refer to them without being tied to any particular logic system, and we also give two concrete instantiations of these concepts using Classical Propositional Logic on one side, and DeLP on the other. In section 4 a basic set of desirable properties for an abstract collaborative dialogue model is proposed. To this end, the notion of relevance is discussed, defining two levels of relevance in dialogues. Illustrative examples using DeLP are provided. Finally, in section 5 we comment on some existing works in the area, and in section 6 we summarize the main contributions of this work, as well as pointing out some issues that have been left for further research in future works.

2 DELP OVERVIEW

Defeasible Logic Programming (DeLP) is a formalism which combines results of Logic Programming and Defeasible Argumentation. It has the declarative capability of representing weak information in the form of *defeasible rules*, and a defeasible argumentation inference mechanism for warranting the entailed conclusions. A brief explanation is included below (see [8] for full details).

2.1 The Language

A *defeasible logic program* (*de.l.p.*) $\mathcal{P}$ is a set of facts, strict rules and defeasible rules, defined as follows. Facts are ground literals representing atomic information, or the negation of atomic information using the strong negation "$\sim$" (e. g. $bird(tweety)$ or $\sim flies(tweety)$). *Strict Rules* represent non-defeasible information and are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (e. g. $bird \leftarrow penguin$ or $\sim innocent \leftarrow guilty$). *Defeasible Rules* represent tentative information that may be used if nothing could be posed against it, and are denoted $L_0 \prec L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals (e. g. $flies \prec bird$ or $\sim flies \prec bird, broken\_wing$). Observe that strict and defeasible rules are ground. However, following the usual convention, some examples use "schematic rules" with variables. To distinguish variables, as usual, they start with an uppercase letter.

When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules. From a program (Π, Δ) contradictory literals could be derived, since *Strong negation* is allowed in the head of rules. Nevertheless, the set Π (which is used to represent non-defeasible information) must possess certain internal coherence. Therefore, no pair of contradictory literals can be derived from Π.

2.2 The Inference Mechanism

For the treatment of contradictory knowledge, DeLP incorporates a defeasible argumentation formalism. This formalism allows the identification of the pieces of knowledge that are in contradiction, and a *dialectical process* is used for deciding which information prevails as warranted. This dialectical process involves the construction and evaluation of arguments that either support or interfere with the query under analysis.

In short, an *argument* for a literal L, denoted $\langle \mathcal{A}, L \rangle$, is a minimal set $\mathcal{A}$ of defeasible rules, ($\mathcal{A} \subseteq \Delta$), such that $\mathcal{A} \cup \Pi$ is non-contradictory and there is a derivation for L from $\mathcal{A} \cup \Pi$. A literal L is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting L. To establish if $\langle \mathcal{A}, L \rangle$ is a non-defeated argument, *defeaters* for $\langle \mathcal{A}, L \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, L \rangle$. In the examples in this paper we assume that the comparison criterion is *generalized specificity* (see [8]). Since defeaters are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *argumentation line* is constructed, where each argument defeats its predecessor. To avoid undesirable sequences, that may represent circular or fallacious argumentation lines, in DeLP an argumentation line is *acceptable* if it satisfies certain constraints (see [8]). Clearly, there might be more than one defeater for a particular argument. Therefore, many acceptable argumentation lines could arise from one argument, leading to a tree structure. In a *dialectical tree*, every node (except the root) represents a defeater of its parent, and leaves correspond to non-defeated arguments. A dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is defeated. Given a literal h and an argument $\langle \mathcal{A}, h \rangle$, every node in the dialectical tree $\mathcal{T}\langle \mathcal{A}, h \rangle$ is recursively marked as "D" (*defeated*) or "U" (*undefeated*), obtaining a marked dialectical tree $\mathcal{T}^*\langle \mathcal{A}, h \rangle$ (see [8] for a detailed explanation of the marking procedure). If the root of $\mathcal{T}^*\langle \mathcal{A}, h \rangle$ is marked as "U", then we will say that $\mathcal{T}^*\langle \mathcal{A}, h \rangle$ *warrants* h and that h is *warranted* from $\mathcal{P}$. There are four possible answers for a query h: YES if h is warranted, NO if $\sim h$ is warranted, UNDECIDED if neither h nor $\sim h$ are warranted, and UNKNOWN if h is not in the language of the program.

Example 1 *Consider the following de.l.p. and the query $h = flies(tweety)$.*

$$\mathcal{P} = \left\{ \begin{array}{ll} bird(X) \leftarrow penguin(X) & flies(X) \prec bird(X) \\ penguin(tweety) & \sim flies(X) \prec penguin(X) \end{array} \right\}$$

An argument for literal h is $\mathcal{A}_1 = \{flies(X) \prec bird(X)\}$. An argument for $\sim h$ is $\mathcal{A}_2 = \{\sim flies(X) \prec penguin(X)\}$. $\mathcal{A}_2$ is a defeater for $\mathcal{A}_1$ (using generalized specificity). $\sim h$ results warranted after the dialectical process, while h results not warranted. The answer to the query is NO.

3 PRELIMINARY DEFINITIONS

We assume the existence of three logical languages: the *Knowledge Representation Language* $\mathcal{L}$, the *Query Language* $\mathcal{L}_{Query}$, and the *Answer Language* $\mathcal{L}_{Answer}$. We also assume that $\mathcal{L} = \mathcal{L}_{Facts} \cup \mathcal{L}_{Rules}$, with $\mathcal{L}_{Facts} \cap \mathcal{L}_{Rules} = \emptyset$. On top of these three languages, abstract operators for *inference* and *abduction* are defined. The inference operator allows to answer queries based on an individual set of knowledge. The abduction operator, defined in terms of the former, allows to discover what information is missing for inferring certain conclusion (that is, what are the minimal sets of facts that could be added so that the inference operator obtains a predetermined answer to a query). Since different agents may have contradictory information, a *consistency* notion and a *consolidation* operator (for resolving the inconsistency) are also abstractly defined. It is important to mention that the inference operator will be always applied to consistent sets.

Definition 1 (Consistency) *A* consistency notion *is a function* Consistent: $2^{\mathcal{L}} \Rightarrow \{true, false\}$.

Definition 2 (Consolidation) *A* consolidation operator *is a function* $! : 2^{\mathcal{L}} \Rightarrow 2^{\mathcal{L}}$ *such that (al least) the following conditions hold for all* $\mathcal{K} \subseteq \mathcal{L}$*:*

- Inclusion*:* $\mathcal{K}! \subseteq \mathcal{K}$.
- Consistency*:* Consistent$(\mathcal{K}!) = true$.
- Vacuity*: if (*Consistent$(\mathcal{K}) = true$*) then* $\mathcal{K}! = \mathcal{K}$.

It is worth mentioning that we are not concerned here with technical details of the consolidation mechanism. We will just assume that there exists a function which returns a consistent subset of the original inconsistent one, so that the inference operator can be applied. Other properties, regarding the idea of *minimal change*, are usually required from a consolidation operator, but these are out of the scope of this particular work. Further information on these topics can be found in the *belief revision* literature ([1], [9], [10] and [11], among others).

Definition 3 (Inference) *An* inference operator *is a function* Infer: $2^{\mathcal{L}} \times \mathcal{L}_{Query} \Rightarrow \mathcal{L}_{Answer}$*, such that (at least) the following condition holds for all* $\mathcal{K} \subseteq \mathcal{L}$ *and* $Q \in \mathcal{L}_{Query}$*:*

- Compactness*: if* Infer$(\mathcal{K}, Q) = \mathcal{A}$ *then there exists a finite subset* $\mathcal{S} \subseteq \mathcal{K}$ *such that* Infer$(\mathcal{S}, Q) = \mathcal{A}$.

As mentioned above, the inference function is defined over consistent sets of knowledge. Thus, it will be always applied over consolidated sets. However, the consolidation step will be omitted, for simplicity reasons, when it is clear that the regarding set is already consistent.

Definition 4 (Abduction) *An* abduction operator *is a function* Abduce: $2^{\mathcal{L}} \times \mathcal{L}_{Query} \times \mathcal{L}_{Answer} \Rightarrow 2^{2^{\mathcal{L}_{Facts}}}$*, such that* $\mathcal{X} \in$ Abduce$(\mathcal{K}, Q, \mathcal{A})$ *if and only if:*

1. Infer$((\mathcal{K} \cup \mathcal{X})!, Q) = \mathcal{A}$*, and*
2. $\nexists\ \mathcal{S} \subset \mathcal{X}$ *such that* Infer$((\mathcal{K} \cup \mathcal{S})!, Q) = \mathcal{A}$.

Two concrete instantiations of these concepts are given next. In example 2, Classical Propositional Logic is used for representing knowledge and reasoning. In example 3, the DeLP formalism is used. All subsequent examples in this paper use the instantiation of languages and operators provided in example 3. The reason we have chosen DeLP for exemplifying the concepts introduced in this paper, is that this formalism allows for interesting and, at the same time, simple examples.

Example 2 *Let* $\mathcal{L}$ *be the language of Classical Propositional Logic (PL),* $\mathcal{L}_{Query}$ *the set of atomic formulas in PL, and* $\mathcal{L}_{Answer}$ *the set* {true, false, undetermined}*. The* consistency notion *is defined as usual:* Consistent*(*$\mathcal{K}$*) iff* $\mathcal{K} \nvdash \bot$*. The* inference function *is defined as usual:* Infer*(*$\mathcal{K}$*,* α*) is* true *if* $\mathcal{K} \vdash \alpha$, false *if* $\mathcal{K} \vdash \sim\alpha$*, and* undetermined *otherwise (i.e.* $\mathcal{K} \nvdash \alpha$ *and* $\mathcal{K} \nvdash \sim\alpha$*). The* consolidation function *considered in this example returns the intersection of all the maximal consistent subsets of* $\mathcal{K}$ [1]*. For instance, the set* $\{a, \sim a\}$ *has two maximal consistent subsets:* $\{a\}$ *and* $\{\sim a\}$*, whose intersection is the empty set. The* abduction operator *is constructed according to definition 4, based on the inference and consolidation functions. Some representative examples showing the behavior of this last operator are given next. Note that in some cases the result is a unitary set containing the empty set, while in others the result itself is the empty set.*

- $\mathcal{K}_1 = \{b \rightarrow a,\ c \wedge d \rightarrow b\}$
  - Abduce*(*$\mathcal{K}_1$*, a, true)*= $\{\{a\}, \{b\}, \{c, d\}\}$
- $\mathcal{K}_2 = \{a,\ b \rightarrow a\}$
  - Abduce*(*$\mathcal{K}_2$*, a, true)*= $\{\emptyset\}$
- $\mathcal{K}_3 = \{\sim a\}$
  - Abduce*(*$\mathcal{K}_3$*, a, false)*= $\{\emptyset\}$
  - Abduce*(*$\mathcal{K}_3$*, a, undetermined)*= $\{\{a\}\}$
  - Abduce*(*$\mathcal{K}_3$*, a, true)*= $\emptyset$

Example 3 *Let* $\mathcal{L}$ *be the DeLP language,* $\mathcal{L}_{Query}$ *the set of literals of the DeLP language, and* $\mathcal{L}_{Answer}$ *the set* {Yes, No, Undecided, Unknown}*. The* consistency notion *and the* consolidation function *are defined in the same way as in example 2, but over the set* Π *of strict rules and facts:* Consistent*(*(Π, Δ)*) iff* $\Pi \nvdash \bot$*, and (*Π_1*,* Δ*)! = (*Π_2*,* Δ*) where* Π_2 *is the intersection of all the maximal consistent subsets of* Π_1*. The* inference function *is defined as explained in section 2:* Infer*(*$\mathcal{P}$*, h) returns* Yes *if h is warranted,* No *if* $\sim h$ *is warranted,* Undecided *if neither h nor* $\sim h$ *are warranted, and* Unknown *if h is not in the language of the program. The behavior of the resulting* abduction operator *is illustrated by the following example.*

- $\mathcal{K} = \{\ bird(X) \leftarrow penguin(X),\ \sim flies(X) \prec penguin(X),\ flies(X) \prec bird(X)\ \}$
  - Abduce$(\mathcal{K}, flies(tweety), Yes) = \{\{flies(tweety)\},\ \{bird(tweety)\}\}$
  - Abduce$(\mathcal{K}, flies(tweety), No) = \{\{\sim flies(tweety)\},\ \{penguin(tweety)\}\}$

4 PROPERTIES FOR COLLABORATIVE DIALOGUES

A *collaborative dialogue* is one in which all the participants have the same goal: to obtain a unified conclusion about the topic under discussion. Collaborative agents have no preferences for particular conclusions, so they share any knowledge considered relevant to the progress of the dialogue. Next, an abstract dialogue operator is defined, under the following assumptions: (1) all agents use the same language for representing knowledge, and (2) each single move in the dialogue consists in an agent publishing a subset of its private knowledge base.

Definition 5 (Collaborative Dialogue) *A* collaborative dialogue operator *is a non-deterministic function* C-dialogue*:* $(2^{\mathcal{L}})^n \times \mathcal{L}_{Query} \Rightarrow \mathcal{L}_{Answer} \times (2^{\mathcal{L}})^m \times \mathcal{N}^m$*, where* $2^{\mathcal{L}}$ *is the powerset of* $\mathcal{L}$*,* $\mathcal{N}$ *is the set of natural numbers,* $n \in \mathcal{N}$ *is the number of agents involved in the dialogue, and* $m \in \mathcal{N}$ *is the number of steps (or moves) of the resulting dialogue.*

[1]Different consolidation functions, satisfying the conditions of definition 2, could be defined. The one chosen here is called *full meet* ([1]) consolidation.

The collaborative dialogue operator takes a tuple ($\mathcal{K}_1 \ldots \mathcal{K}_n$, Q) and returns a tuple ($\mathcal{A}$, $\mathcal{X}_1 \ldots \mathcal{X}_m$, $i_1 \ldots i_m$), where: $\mathcal{K}_1 \ldots \mathcal{K}_n$ are the private knowledge bases of the agents (these are assumed to be consistent), Q is the query (or topic) of the dialogue, $\mathcal{A}$ is the outcome of the dialogue, and $\mathcal{X}_1 \ldots \mathcal{X}_m$ along with $i_1 \ldots i_m$ represent the sequence of moves made by the agents. Each move is a pair $< \mathcal{X}, i >$, meaning that agent i, with knowledge base $\mathcal{K}_i$, said $\mathcal{X}$. As mentioned above, we assume that $\mathcal{X} \subseteq \mathcal{K}_i$ for every move $< \mathcal{X}, i >$. The non-determinism captures the idea of several possible moves at each step of the dialogue, and the model not specifying a particular one. The choices may involve different agents, as well as different utterances by the same agent.

4.1 The Notion of Relevance

The *relevance* of knowledge is a key notion when studying properties of dialogue models. Hence, we will define it accurately before enunciating any property. The difficulty that emerges when determining relevance within a context of incomplete information, such as a dialogue (where each agent has to discover relevant subsets of its own knowledge, but ignores what the others know), was already addressed in section 1. Suppose that, in a first approach, we adopt the following notion of relevance: *"a contribution made by an agent in a dialogue is relevant if it changes the current conclusion about the topic at issue"*. It seems reasonable and it works, for example, in the following case: agents A_1 and A_2 engage in dialogue to determine whether tweety flies or not, A_1 knows that all birds fly and that tweety is a bird, and A_2 knows that tweety is a penguin and penguins are a special kind of bird which does not fly. The knowledge of agents A_1 and A_2 is represented using DeLP as follows: $\mathcal{K}_1 = \{flies(X) \prec bird(X),\ bird(tweety)\}$ and $\mathcal{K}_2 = \{penguin(tweety),\ bird(X) \leftarrow penguin(X),\ {\sim}flies(X) \prec penguin(X)\}$. In this case, the knowledge of A_1 can change the current conclusion at the initial step of the dialogue (*Unknown* to *Yes*), and after that is said, the knowledge of A_2 can change again the current conclusion (*Yes* to *No*). However, consider now this other situation: A_1 only knows that penguins do not fly and A_2 only knows that tweety is a penguin, which is represented in DeLP as follows: $\mathcal{K}_1 = \{{\sim}flies(X) \prec penguin(X)\}$ and $\mathcal{K}_2 = \{penguin(tweety)\}$. In this case both agents know relevant things, though none of them can, by itself, change the current conclusion.

A more sophisticated notion of relevance is needed. More precisely, we need to model not only when something is *directly relevant* (that is, when something actually changes the current conclusion), but also when something is *potentially relevant* (that is, when something might help to change the current conclusion in a further step in the dialogue). We capture this idea of *potential relevance* as follows: *"a contribution made by an agent in a dialogue is potentially relevant if there* might *exist some knowledge (facts) of some other agent(s) such that they might together change the current conclusion"*. In other words, a contribution is potentially relevant if it changes the set of facts that are missing for changing the current conclusion. We restrict the missing information to be only facts (no rules) because otherwise *any* knowledge would be considered potentially relevant.

The following terminology formalizes the previous ideas, and then an illustrative example using the DeLP formalism is provided. Given $\mathcal{K} \subseteq \mathcal{L}$, $\mathcal{X} \subseteq \mathcal{L}$ and $Q \in \mathcal{L}_{Query}$, we define the conditions under which the set $\mathcal{X}$ is considered a relevant contribution to the set $\mathcal{K}$, in the context of a dialogue about Q. We assume underlying inference (*Infer*) and consolidation (!) mechanisms.

Definition 6 (Direct Q-Relevant Contribution) *Given $\mathcal{K} \subseteq \mathcal{L}$ and $Q \in \mathcal{L}_{Query}$, we say that $\mathcal{X} \subseteq \mathcal{L}$ is a* direct Q-relevant contribution *to $\mathcal{K}$ if and only if* Infer*($\mathcal{K}!$, Q)* $\neq$ Infer*(($\mathcal{K} \cup \mathcal{X}$)!,* *Q).*

Definition 7 (Potential Q-Relevant Contribution) *Given $\mathcal{K} \subseteq \mathcal{L}$ and $Q \in \mathcal{L}_{Query}$, we say that $\mathcal{X} \subseteq \mathcal{L}$ is a* potential Q-relevant contribution *to $\mathcal{K}$ if and only if* Abduce*($\mathcal{K}$, Q, $\mathcal{A}$)* $\neq$ Abduce*($\mathcal{K} \cup \mathcal{X}$, Q, $\mathcal{A}$) for some $\mathcal{A} \in \mathcal{L}_{Answer}$.*

Definition 8 (Q-Relevant Contribution) *Given $\mathcal{K} \subseteq \mathcal{L}$ and $Q \in \mathcal{L}_{Query}$, we say that $\mathcal{X} \subseteq \mathcal{L}$ is a* Q-relevant contribution *to $\mathcal{K}$ if and only if $\mathcal{X}$ is either a direct or potential Q-relevant contribution to $\mathcal{K}$.*

Example 4 *Consider the set $\mathcal{K} = \{flies(X) \prec bird(X),\ bird(X) \leftarrow penguin(X)\}$ and the query $Q = flies(tweety)$, where* Infer*($\mathcal{K}$, Q) =* Unknown. *Some Direct Q-relevant contributions to $\mathcal{K}$ are, for instance: $\mathcal{X}_1 = \{bird(tweety)\}$ and $\mathcal{X}_2 = \{\sim flies(X) \prec penguin(X), penguin(tweety)\}$, where $\mathcal{X}_1$ changes the conclusion from* Unknown *to* Yes*, and $\mathcal{X}_2$ changes the conclusion from* Unknown *to* No. *Besides, we could think of some Potential Q-relevant contributions to $\mathcal{K}$, for instance: $\mathcal{X}_3 = \{\sim flies(X) \prec penguin(X)\}$, $\mathcal{X}_4 = \{flies(X) \prec plane(X)\}$ and $\mathcal{X}_5 = \{bird(X) \prec chicken(X)\}$, where $\mathcal{X}_3$ adds a potential reason for answering* No *(the element $\{penguin(tweety)\}$ is added to* Abduce*($\mathcal{K}$, Q,* No*)), and the other two sets add potential reasons for answering* Yes *(the element $\{plane(tweety)\}$ in the case of $\mathcal{X}_4$, and $\{chicken(tweety)\}$ in the case of $\mathcal{X}_5$, is added to* Abduce*($\mathcal{K}$, Q,*Yes*)).*

Observe that the relevance notion is *non-monotonic*, i.e.: a set which is not a Q-relevant contribution to $\mathcal{K}$ may contain a Q-relevant contribution to $\mathcal{K}$. This will be reminded later, in relation to the completeness property for collaborative dialogues.

4.2 Dialogue Properties

Having specified the formal meaning of relevance, we are now able to define some desirable properties for collaborative dialogues. As anticipated in section 1, the properties point to: (1) a reasonable connection between the steps and the conclusion, (2) the relevance of each step, (3) the completeness of the sequence of steps, and (4) the finiteness of the sequence of steps. It is assumed, as in section 4.1, that there are inference and consolidation operators associated to the dialogue (*Infer* and !, respectively). The formal definitions will be introduced first, and then a more detailed explanation of each property will be given, as well as an example dialogue that violates it. The examples are written using the DeLP formalism, and consolidations are omitted for simplicity when the set under consideration is already consistent. Let *C-dialogue* be a collaborative dialogue operator, then it is desirable that the following conditions hold for every tuples $s = (\mathcal{K}_1 \ldots \mathcal{K}_n, Q)$ and $t = (\mathcal{A}, \mathcal{X}_1 \ldots \mathcal{X}_m, i_1 \ldots i_m)$ such that t is a possible result of applying *C-dialogue* to s.

1. *Correctness: Infer*$((\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_m)!\ , Q) = \mathcal{A}$.
2. *Relevance:* $\forall_{1 \leq j \leq m}$, $\mathcal{X}_j$ is a Q-relevant contribution for $\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_{j-1}$.
3. *Termination:* $i_1 \ldots i_m$ is a finite sequence of natural numbers.
4. *Completeness:* $\forall_{1 \leq i \leq n} : \mathcal{K}_i$ is not a Q-relevant contribution for $\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_m$.

The *Correctness* property states that the outcome of the dialogue should follow from all what the agents have said. It prevents situations like the following: suppose that agents A_1 and A_2 want to determine whether *tweety* flies or not. A_1 says that all birds fly, A_2 says that *tweety* is a bird, and then they conclude that *tweety* does not fly, which is clearly not the expected conclusion. Example 5 illustrates this situation using the DeLP formalism.

Example 5 *Consider the tuples s= ($\mathcal{K}_1$, $\mathcal{K}_2$, Q) and t= ($\mathcal{A}$, $\mathcal{X}_1$, $\mathcal{X}_2$, 1, 2) instantiated below. It can be seen in this example that the correctness property is violated, since* Infer*($\mathcal{X}_1 \cup \mathcal{X}_2$, Q)* = Yes *but the conclusion achieved ($\mathcal{A}$) is* No.

$$\begin{array}{lcl} \mathcal{K}_1 = \{flies(X) \prec bird(X)\} & & \mathcal{X}_1\ (A_1\ says) = \{flies(X) \prec bird(X)\} \\ \mathcal{K}_2 = \{bird(tweety)\} & \Rightarrow & \mathcal{X}_2\ (A_2\ says) = \{bird(tweety)\} \\ Q = flies(tweety) & & \mathcal{A} = \text{No} \end{array}$$

The *Relevance* property prevents the agents from saying something which is not related to the topic of the dialogue, or which is not influential in the determination of the conclusion. Taking the previous example, it prevents situations like the following: A_1 says that all birds fly and then A_2 says that it is raining outside. Example 6 illustrates this situation using the DeLP formalism.

Example 6 *Consider the tuples s= ($\mathcal{K}_1$, $\mathcal{K}_2$, Q) and t= ($\mathcal{A}$, $\mathcal{X}_1$, $\mathcal{X}_2$, 1, 2) instantiated below. It can be seen in this example that the relevance property is violated, since $\mathcal{X}_2$ is not a Q-relevant contribution to $\mathcal{X}_1$ (*Infer*($\mathcal{X}_1$, Q) =* Infer*($\mathcal{X}_1 \cup \mathcal{X}_2$, Q), and* Abduce*($\mathcal{X}_1$, Q, Ans) =* Abduce*($\mathcal{X}_1 \cup \mathcal{X}_2$, Q, Ans) for any element of $\mathcal{L}_{Answer}$).*

$$\begin{array}{lcl} \mathcal{K}_1 = \{flies(X) \prec bird(X)\} & & \mathcal{X}_1\ (A_1\ says) = \{flies(X) \prec bird(X)\} \\ \mathcal{K}_2 = \{raining\_outside\} & \Rightarrow & \mathcal{X}_2\ (A_2\ says) = \{raning\_outside\} \\ Q = flies(tweety) & & \mathcal{A} = \text{Unknown} \end{array}$$

The *Completeness* property prevents the agents from not saying something that is relevant for the determination of the conclusion. Notice that the whole knowledge base of each agent is required to be non-relevant at the end of the dialogue (in contrast with requiring the bases not to contain relevant subsets). The reason is that, as remarked in section 4.1, an agent's knowledge base may contain a Q-relevant contribution without being relevant itself. The completeness property prevents situations like the following: A_1 says that all birds fly, A_2 says that *tweety* is a bird (specifically a penguin, which is a kind of bird) and then the dialogue terminates. However, A_2 knows that penguins are a special kind of bird which does not fly. Example 7 illustrates this situation using the DeLP formalism, and example 8 illustrates a similar situation where the incompleteness of the dialogue is not so evident to each agent.

Example 7 *Consider the tuples s= ($\mathcal{K}_1$, $\mathcal{K}_2$, Q) and t= ($\mathcal{A}$, $\mathcal{X}_1$, $\mathcal{X}_2$, 1, 2) instantiated below. It can be seen in this example that the completeness property is violated, since $\mathcal{K}_2$ is a direct Q-relevant contribution to $\mathcal{X}_1 \cup \mathcal{X}_2$ (adding the rule $\sim flies(X) \prec penguin(X)$ would change the conclusion from* Yes *to* No*).*

$$\begin{array}{lcl} \mathcal{K}_1 = \{flies(X) \prec bird(X)\} & & \mathcal{X}_1\ (A_1\ says) = \{flies(X) \prec bird(X)\} \\ \mathcal{K}_2 = \left\{ \begin{array}{l} bird(X) \leftarrow penguin(X), \\ penguin(tweety) \\ \sim flies(X) \prec penguin(X) \end{array} \right\} & \Rightarrow & \mathcal{X}_2\ (A_2\ says) = \left\{ \begin{array}{l} bird(X) \leftarrow penguin(X), \\ penguin(tweety) \end{array} \right\} \\ Q = flies(tweety) & & \mathcal{A} = \text{Yes} \end{array}$$

Example 8 *Consider the tuples s= ($\mathcal{K}_1$, $\mathcal{K}_2$, Q) and t= ($\mathcal{A}$, $\mathcal{X}_1$, $\mathcal{X}_2$, 1, 2) instantiated below. It can be seen in this example that the completeness property is violated, since $\mathcal{K}_1$ is a potential Q-relevant contribution to $\mathcal{X}_1 \cup \mathcal{X}_2$.*

$$\mathcal{K}_1 = \left\{ \begin{array}{l} flies(X) \prec bird(X), \\ \sim flies(X) \prec bird(X), broken\_wing(X) \end{array} \right\} \qquad \mathcal{X}_1 \ (A_1 \ says) = \{flies(X) \prec bird(X)\}$$

$$\mathcal{K}_2 = \left\{ \begin{array}{l} bird(tweety), \\ broken\_wing(tweety) \end{array} \right\} \qquad \Rightarrow \qquad \mathcal{X}_2 \ (A_2 \ says) = \{ \ bird(tweety) \ \}$$

$$Q = flies(tweety) \qquad \mathcal{A} = \text{Yes}$$

Finally, the *Termination* property states that every dialogue should terminate after a finite sequence of steps. It prevents situations like the one illustrated in example 9. Observe that the dialogue of example 9 does not satisfy relevance either (only the first two steps are relevant). In fact, since we assume that agents publish subsets of their private knowledge, it can be proven that Termination is implied by Relevance.

Example 9 *Consider the tuples* $s = (\mathcal{K}_1, \mathcal{K}_2, Q)$ *and* $t = (\mathcal{A}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \ldots 1, 2, 1, \ldots)$ *instantiated below. It can be seen in this example that the termination property is violated, since tuple t contains an infinite sequence of steps. Also notice that for all* $j \geq 3$*:* $\mathcal{X}_j$ *is not a Q-relevant contribution to* $\mathcal{X}_1 \cup \ldots \cup \mathcal{X}_{j-1}$*, so the relevance property is also violated.*

$$\begin{array}{lll} \mathcal{K}_1 = \{flies(tweety)\} & & \mathcal{X}_1 \ (A_1 \ says) = \{flies(tweety)\} \\ \mathcal{K}_2 = \{\sim flies(tweety)\} & \Rightarrow & \mathcal{X}_2 \ (A_2 \ says) = \{\sim flies(tweety)\} \\ Q = flies(tweety) & & \mathcal{X}_3 \ (A_1 \ says) = \{flies(tweety)\} \\ & & \vdots \\ & & \mathcal{X}_j \ (A_2 \ says) = \{\sim flies(tweety)\} \\ & & \mathcal{X}_{j+1} \ (A_1 \ says) = \{flies(tweety)\} \\ & & \vdots \end{array}$$

5 RELATED WORK

There are in the literature a variety of works that investigate dialogues from different perspectives. Most of them propose a formal *model* for some type of dialogue, and identify some properties of the dialogues generated by their system (these include [3], [6], [4], [12], [5] and [7]). In [3], the authors propose a general framework for argumentation-based *negotiation*, and study some properties of the outcomes: *termination*, and what they called *completeness* and *soundness* (although the motivation may be similar, the concept captured by these last two properties differs from the one captured by our properties of *completeness* and *correctness*). In [6], a formal framework for argumentation-based dialogue between autonomous agents which are looking for a common agreement about a collective choice, is proposed. Properties of the framework are studied (*termination* and *optimal outcome*). In [4], [12] and [5] together, the authors study argumentation-based dialogues between agents, focusing in dialogues *over beliefs* (namely: *inquiry*, *persuasion* and *information-seeking*). They define a set of locutions by which agents can trade arguments, a set of agent attitudes which relate what arguments an agent can build and what locutions it can make, and a set of protocols by which dialogues can be carried out. They consider some properties of dialogues under the protocols (in particular *termination* and *complexity*), and then extend their work by examining the outcomes of the dialogues, and investigating the extent to which outcomes are dependent on tactical play by the agents. In [7], an *inquiry* dialogue protocol, based on argumentation, is defined. A strategy that selects one of the legal moves to make is also defined. The authors propose a benchmark against which they compare the dialogues, being the arguments that can be constructed from the union of the agents beliefs, and use this to define *soundness* and *completeness* properties for inquiry dialogues. This completeness property captures almost the same idea as ours, although it is enunciated for a system constructed on the basis of a particular reasoning (argumentative) model. Besides, our way of designing that property is more *constructive*, in the sense that it gives an idea on how to construct a dialogue model which satisfies it.

There is a minority of works, including [2] and [13], which focus on *properties* of dialogues in general, without proposing any particular protocol. In [2] different measures for analyzing *persuasion* dialogs, from the point of view of an external agent, are defined. In particular, three kinds of measures are proposed: measures of the quality of the exchanged arguments, measures of the behavior of each agent (in terms of its coherence, aggressiveness, and novelty), and measures of the quality of the dialog itself in terms of the relevance and usefulness of its moves. In [13], the authors investigate the relevance of utterances in an argumentation based dialogue. They study three notions of relevance, and show how they can affect the outcome of the dialogue.

6 CONCLUSIONS AND FUTURE WORK

In this work, we have presented a set of common sense properties (namely: termination, correctness, relevance and completeness) that should be satisfied by formal models of collaborative dialogue, showing in this way that it is possible to identify, and formally define, these properties without specifying a particular underlying logic or a particular dialogue protocol. To that end, we have also introduced an abstract notion of relevance in dialogue, which is able to handle the problem of the incompleteness of knowledge. Some assumptions have been made over the nature of the dialogues being considered here: (1) agents can assert only subsets of their private knowledge, (2) agents share a common knowledge representation language and inference mechanism, and (3) agents share a common criterion for resolving inconsistencies (i.e. there is a unified consolidation function).

Some issues have been left for further investigation in future works. First, we aim at relaxing some of the assumptions made here, for instance allowing agents to disagree about the consolidation criterion. Second, and considering the constructive nature of the properties presented here, we aim at investigating efficient implementations of models that satisfy them (this is mainly concerned with methods of finding relevant subsets of knowledge). Last, we plan to extend this work to non-collaborative dialogue types, such as negotiation and persuasion.

REFERENCES

[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[2] Leila Amgoud and Florence Dupin de Saint-Cyr. Measures for persuasion dialogs: A preliminary investigation. *In 2nd International Conference on Computational Models of Argument, COMMA'08, Toulouse - France*, May 2008.

[3] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. A unified and general framework for argumentation based negotiation. *In 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Honolulu, Hawai'i*, May 2007.

[4] Leila Amgoud, Nicolas Maudet, and Simon Parsons. Modelling dialogues using argumentation. *In 4th International Conference on MultiAgent Systems, ICMAS'2000, Boston, USA. IEEE Press*, 2000. July.

[5] Leila Amgoud, Simon Parsons, and Michael Wooldridge. On the formal outcomes of formal inter-agent dialogues. *In 2nd International joint conference on Autonomous Agents and Multi-Agent systems, AAMAS'03, Melbourne, Australie. ACM Press*, pages 616–623, July 2003.

[6] Leila Amgoud, Henri Prade, and Sihem Belabbes. Towards a formal framework for the search of a consensus between autonomous agents. *In the proceedings of the 4th International joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS'2005, Utrecht. Franck Dignum, Michael Wooldridge, Sven Koenig, Sarit Kraus (Eds.), ACM Press*, pages 537–543, July 2005.

[7] Elizabeth Black and Anthony Hunter. A generative inquiry dialogue system. *In 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07), Honolulu, Hawai'i*, May 2007.

[8] Alejandro J. García and Guillermo R. Simari. Defeasible logic programming: An argumentative aproach. 2004.

[9] Sven Ove Hansson. *Belief Base Dynamics*. PhD thesis, Uppsala University, Department of Philosophy, Uppsala, Sweden, 1991.

[10] Sven Ove Hansson. Kernel contraction. *The Journal of Symbolic Logic*, 59:845–859, 1994.

[11] Sven Ove Hansson. Semi-revision. *Journal of Applied Non-Classical Logic*, 7:151–175, 1997.

[12] Simon Parsons, Leila Amgoud, and Michael Wooldridge. An analysis of formal inter-agent dialogues. *In 1st International Joint Conference on Autonomous Agents and Multi-Agent systems, AAMAS'2002, Bologna, Italy. ACM Press*, pages 394–401, July 2002.

[13] Simon Parsons, Peter McBurney, Elizabeth Sklar, and Michael Wooldridge. On the relevance of utterances in formal inter-agent dialogues. *In 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07), Honolulu, Hawai'i*, May 2007.

[14] Simon Parsons, Michael Wooldridge, and Leila Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13:347–376, 2003.

[15] Douglas Walton and Erik C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, 1995.