An Argument-based Approach to Using Multiple Ontologies

Elizabeth Black[1], Anthony Hunter[2], and Jeff Z. Pan[3]

[1] Department of Engineering Science,
University of Oxford, Parks Road, Oxford, OX1 3PJ, UK
[2] Department of Computer Science,
University College London, Gower Street, London WC1E 6BT, UK
[3] Department of Computing Science,
University of Aberdeen, Aberdeen AB24 3UE, UK

Abstract. Logic-based argumentation offers an approach to querying and revising multiple ontologies that are inconsistent or incoherent. A common assumption for logic-based argumentation is that an argument is a pair $\langle \Phi, \alpha \rangle$ where Φ is a minimal subset of the knowledgebase such that Φ is consistent and Φ entails the claim α. Using dialogue games, agents (each with its own ontology) can exchange arguments and counterarguments concerning formulae of interest. In this paper, we present a novel framework for logic-based argumentation with ontological knowledge. As far as we know, this is the first proposal for argumentation with multiple ontologies via dialogues. It allows two agents to discuss the answer to queries concerning their knowledge (even if it is inconsistent) without one agent having to copy all of their ontology to the other, and without the other agent having to expend time and effort merging that ontology with theirs. Furthermore, it offers the potential for the agents to incrementally improve their knowledge based on the dialogue by checking how it differs from the other agent's.

1 Introduction

Inconsistency and incoherence are recognized as significant problems in managing ontological knowledge (e.g. [11]). These problems are particularly an issue when using multiple ontologies. Current solutions include the "maxcon" approach (which involves merging ontologies by selecting a maximal consistent subset of the union of the multiple ontologies) and the "oracle" approach (which involves constructing a merged consistent ontology by getting extra information to help resolve the conflicts). Unfortunately, the maxcon approach results in a loss of useful information, as it may not be certain which subset to choose, and therefore an arbitrary choice is made, and the oracle approach involves a lot of work that may not be necessary if for example a query can be answered from a small part of the agents' knowledge, and furthermore that this knowledge may not even be in conflict.

To address these problems, here we explore an alternative approach which involves only focusing on the subset of the union of the ontologies that is required for answering queries. Our approach is to keep the ontologies separate, and associate each ontology with an agent. Then, the agents enter into a dialogue in which arguments are exchanged concerning some subject. To simplify our presentation, we restrict consideration to two agents 1 and 2. Each agent i has a personal knowledgebase (or perbase) $\varDelta_i$ that is a description logic ontology (e.g. OWL). We assume each perbase is finite, consistent and coherent, but that normally $\varDelta_1 \cup \varDelta_2$ is inconsistent or incoherent. We also assume they use the same description logic, though not necessarily the same vocabulary (i.e. they can use different names for the same concept and/or the same names for different concepts). Each perbase is private (i.e. the contents of the perbase are only available to the agent). The strategy used by the agent dictates what the agent will make public from its perbase. We also assume a finite knowledgebase of lexical matching knowledge called a lexbase $\varPi$ that contains just formulae of the form $A \sqsubseteq B$ where A and B are named concepts. Each formula in $\varPi$ is obtained by a lexical matching algorithm, and there is some reasonable probability that it is correct. $\varPi$ is public knowledge (i.e. the contents are available to both agents) and $\varPi$ may be inconsistent with either of the perbases. We assume some external agent has decided what formulae are in $\varPi$, and that each agent can choose to draw on formulae from the lexbase as required.

The aim of each agent is to co-operate with the other to answer questions about the ontological information they have, even though there may be conflicts between their perbases. This will allow agents to efficiently and effectively use their own ontology with the other agent's knowledge even when there is conflict between them. A further benefit (which we consider in the discussion) is that each agent can improve their own perbase through the argumentation process, and so undertake a form of partial merging of the other agent's ontological knowledge with their own ontological knowledge.

In the rest of this paper, we consider arguments based on description logic, and present a general framework for dialogical argumentation. We show that using dialogical argumentation allows for agents to use multiple ontologies without having to distribute each ontology to every other agent, and without having to merge the ontological knowledge.

2 Logical Arguments

The usual paradigm for **logic-based argumentation** is that there is a large repository of information, represented by $\varDelta$, from which logical arguments can be constructed for and against arbitrary claims (e.g. [15, 4, 1, 9, 7]). There is no *a priori* restriction on the contents, and the pieces of information in the repository can be arbitrarily complex. Therefore, $\varDelta$ is not expected to be consistent. It need not even be the case that every single formula in $\varDelta$ is consistent.

Our framework adopts a very common intuitive notion of a logical argument. Essentially, an argument is a set of formulae that can be used to prove some claim, together with that claim. Each claim is represented by a formula. Provability is represented by a consequence relation that may be for a logic such as classical logic, or a description logic.

Here we focus on **description logics** (DLs) [2], which is a family of logic-based knowledge representation formalisms used as the underpinning of the standard OWL Web ontology language. DLs are characterised by the constructors (such as $C \sqcap D$, $\exists R.C, \forall R.C$) for building complex concept descriptions, such as $\texttt{Animal} \sqcap \forall \texttt{Eat}.\texttt{Plant}$

(animals that eat only plants), and role descriptions. A DL ontology $\mathcal{O}$ can consist of concept axioms, such as concept inclusion axioms $C \sqsubseteq D$ (e.g., $\texttt{Cow} \sqsubseteq \texttt{Animal} \sqcap \forall\texttt{Eat.Plant}$), role axioms, such as transitive role axioms, and individual axioms, such as concept assertions (e.g. $\texttt{Cow}(\texttt{daisy})$ and $\texttt{Plant}(\texttt{grass1})$), $C(a)$ and role assertions $R(a, b)$ (e.g. $\texttt{Eat}(\texttt{daisy}, \texttt{grass1})$). We use $A \equiv B$ as shorthand for $A \sqsubseteq B$ and $B \sqsubseteq A$. A DL ontology $\mathcal{O}$ is **consistent** if there exists some interpretation $\mathcal{J}$ that satisfies all its axioms. A concept C is **satisfiable** w.r.t. $\mathcal{O}$ if there exists some interpretation $\mathcal{J}$ of $\mathcal{O}$ such that $C^{\mathcal{J}}$ is non-empty. A DL ontology $\mathcal{O}$ is **coherent** if all the named concepts in $\mathcal{O}$ are satisfiable. Negated axioms are closely related to inconsistencies and changes in ontologies. Since well known DLs do not provide enough expressive power to represent negations of all the axioms, we use two relaxed notions of negation proposed in [8], namely consistency-negation and coherence-negation. Intuitively speaking, an axiom α and any of its consistency-negation (coherence-negation) β are inconsistent (incoherent, resp.) with each other; i.e., $\{\alpha, \beta\}$ is inconsistent (incoherent, resp.).

We let $\mathcal{L}$ denote a DL and $\vdash$ denote the consequence relation of the DL. We use $\alpha, \beta, \gamma, \ldots$ to denote DL formulae (axioms), $\Delta, \Phi, \Psi, \ldots$ to denote sets of DL formulae, and $\mathsf{Names}(\alpha)$ to denote the set of names for concepts, roles and individuals from which a formula α is composed.

Definition 1. *An* **argument** *is a pair $\langle\Phi, \alpha\rangle$ s.t.: (1) $\Phi \subseteq \Delta$; (2) Φ is consistent and coherent; (3) $\Phi \vdash \alpha$; and (4) there is no $\Phi' \subset \Phi$ s.t. $\Phi' \vdash \alpha$. We say that $\langle\Phi, \alpha\rangle$ is an argument for α. We call α the* **claim** *and Φ the* **support** *of the argument.*

An undercut is a counterargument that directly opposes the support of an argument.

Definition 2. *Let $\langle\Phi, \alpha\rangle$, $\langle\Psi, \beta\rangle$ be arguments. $\langle\Psi, \beta\rangle$ is an* **undercut** *for $\langle\Phi, \alpha\rangle$ iff $\{\beta\} \cup \Phi$ is inconsistent or incoherent.*

Example 1. Let $\Delta = \{A \sqsubseteq B, B \sqsubseteq C, A \sqsubseteq D, A \sqsubseteq \neg C\}$. Some arguments include $\langle\{A \sqsubseteq B, B \sqsubseteq C\}, A \sqsubseteq C\rangle$ and $\langle\{A \sqsubseteq \neg C\}, A \sqsubseteq \neg C\rangle$ which undercut each other.

We can capture an argument

together with its undercuts, and by recursion, undercuts to undercuts, in a tree structure as follows. We assume that X denotes this set of arguments.

Definition 3. *An* **argument tree** *for ϕ is a tree (N, E, X, f) where N is a set of nodes, E is a set of edges, X is a set of arguments, and $f : N \mapsto X$ assigns an argument to each node s.t. (1) the root is assigned with an argument for ϕ (the* **root argument***); (2) for each node n, if m is a child of n, then $f(m)$ is an undercut of $f(n)$; and (3) for each node n, if m is an ancestor of n in the branch, then the support of $f(n)$ is not a subset of the support of $f(m)$.*

The **dialectical principle** (widely adopted in the literature on argumentatation) evaluates each argument as defeated or undefeated: An argument is **undefeated** if all the undercuts for it are defeated, and an argument is **defeated** if there is a undercut to it that is undefeated. Any argument with no undercuts is undefeated. For example, let A_1 be the root argument, A_2 and A_3 be undercuts to A_1, and A_4 be an undercut to A_3. In this case, only A_2 and A_4 are undefeated.

3 Dialogue Framework

The communicative acts in a dialogue are called **moves**. We assume that there are always exactly two agents taking part in a dialogue, each with its own identifier taken from the set $\mathcal{I} = \{1, 2\}$. Each agent takes it in turn to make a move to the other agent. We also refer to agents using the variables x and $\overline{x}$ such that if x is 1 then $\overline{x}$ is 2 and if x is 2 then $\overline{x}$ is 1. The format for moves is shown in Table 1, and the set of all moves meeting the format is denoted $\mathcal{M}$. For a move m, the function Sender returns the agent that made the move.

We now informally explain the different types of move: A query move $\langle x, \mathsf{query}, \alpha\rangle$ starts a dialogue with the topic α; A posit move $\langle x, \mathsf{posit}, \langle \Phi, \alpha\rangle\rangle$ asserts an argument $\langle \Phi, \alpha\rangle$ by x for the topic, or an undercut for a previous posit; A concede move $\langle x, \mathsf{concede}, \alpha\rangle$ asserts agent x will regard α as valid; and a close move $\langle x, \mathsf{close}, \gamma\rangle$ is used when an agent has no other moves it can make. Note, for a posit move $m = \langle x, \mathsf{posit}, \langle \Phi, \alpha\rangle\rangle$, we say that m is a posit move for α.

A dialogue is simply a sequence of moves, each of which is made from one participant to the other. As a dialogue progresses over time, we denote each timepoint by a natural number. Each move is indexed by a timepoint and exactly one move is made at each timepoint.

Definition 4. *A* **dialogue***, denoted D^t, is a sequence of moves $m_1, \ldots, m_t$ involving two agents in $\mathcal{I} = \{1, 2\}$, s.t. (1) m_1 is of the form $\langle x, \mathsf{query}, \gamma\rangle$; (2) $\mathsf{Sender}(m_s) \in \mathcal{I}$ for $1 \leq s \leq t$; and (3) $\mathsf{Sender}(m_s) \neq \mathsf{Sender}(m_{s+1})$ for $1 \leq s < t$. The* **topic** *of the dialogue D^t is returned by $\mathsf{Topic}(D^t)$ (i.e. $\mathsf{Topic}(D^t) = \gamma$). The set of all dialogues is denoted $\mathcal{D}$.*

The first move of a dialogue D^t must always be a query move (condition 1), every move of the dialogue must be made to a participant of the dialogue (condition 2), and the agents take it in turns to make moves (condition 3). In order to terminate a dialogue, two close moves must appear one immediately after the other in the sequence (called a *matched-close*).

Definition 5. *Let D^t be a dialogue s.t. $\mathsf{Topic}(D^t) = \gamma$. We say that m_s $(1 < s \leq t)$, is a* **matched-close for** *D^t iff $m_{s-1} = \langle x, \mathsf{close}, \gamma\rangle$ and $m_s = \langle \overline{x}, \mathsf{close}, \gamma\rangle$. D^t* **terminates at** *t iff m_t is a matched-close for D^t and there does not exist an s s.t. $s < t$ and D^s terminates at s.*

Move	**Format**
query	$\langle x, \mathsf{query}, \gamma\rangle$
posit	$\langle x, \mathsf{posit}, \langle \Phi, \phi\rangle\rangle$
concede	$\langle x, \mathsf{concede}, \gamma\rangle$
close	$\langle x, \mathsf{close}, \gamma\rangle$

Table 1. The move format, where γ is a formula, $\langle \Phi, \phi\rangle$ is an argument and $x \in \{1, 2\}$ is the agent that makes the move.

Example 2. $\Delta_1 = \{\neg C(b)\}$; $\Delta_2 = \{C(b), R(b,a), C \sqsubseteq \forall R.C, D(a)\}$; $\Pi = \{C \equiv D\}$.

$$\begin{array}{l}\langle 1, \mathsf{query}, C(a)\rangle \\ \langle 2, \mathsf{posit}, \langle\{C(b), R(b,a), C \sqsubseteq \forall R.C\}, C(a)\rangle\rangle \\ \langle 1, \mathsf{posit}, \langle\{\neg C(b)\}, \neg C(b)\rangle\rangle \\ \langle 2, \mathsf{concede}, D(a)\rangle \\ \langle 1, \mathsf{concede}, C \equiv D\rangle \\ \langle 2, \mathsf{posit}, \langle\{C \equiv D, D(a)\}, C(a)\rangle\rangle \\ \langle 1, \mathsf{close}, C(a)\rangle \\ \langle 2, \mathsf{close}, C(a)\rangle\end{array}$$

Agent 2 posits an argument for the topic, and agent 1 provides an undercut to it. Then each agent concedes a formula, and agent 2 uses these to posit an argument for the topic.

Since all our examples are terminated dialogues, from now on we will omit the matched-close moves.

We associate a commitment store with a dialogue, and let it grow monotonically over the course of the dialogue: If an agent posits an argument, the support is added to the commitment store; If an agent concedes a formula, it is added to the commitment store. A commitment store is therefore the union of all the supports of all the arguments that have been publicly posited along with all the formulae that have been publicly conceded by the agents so far. For this reason, when constructing an argument, an agent may make use of not only its own perbase but also those from the commitment store.

Definition 6. *A* **commitment store** Σ^t *is* $\emptyset$ *at* $t = 0$*, and for all* $t \geq 1$*, if* $m_t = \langle x, \mathsf{posit}, \langle \Phi, \phi\rangle\rangle$*, then* $\Sigma^t = \Sigma^{t-1} \cup \Phi$*, else if* $m_t = \langle x, \mathsf{concede}, \alpha\rangle$*, then* $\Sigma^t = \Sigma^{t-1} \cup \{\alpha\}$*, otherwise* $\Sigma^t = \Sigma^{t-1}$.

A protocol is a function that returns the set of moves that are legal for an agent to make at a particular point in a particular type of dialogue. Here we give the specific protocol that we require. It takes the dialogue and the identifier of the agent whose turn it is to move, and returns the set of legal moves that the agent may make.

Definition 7. *Let* D^t *be a dialogue s.t.* $\mathsf{Sender}(m_t) = \overline{x}$*, and* $\mathsf{Topic}(D^t) = \gamma$*. The* **protocol** *for agent* x *is a function* $\mathsf{Protocol}_x : \mathcal{D} \mapsto \wp(\mathcal{M})$ *s.t.* $\mathsf{Protocol}_x(D^t)$ *is*

$$P_x^{\mathsf{posit}}(D^t) \cup P_x^{\mathsf{concede}}(D^t) \cup \{\langle x, \mathsf{close}, \gamma\rangle\}$$

where $P_x^{\mathsf{posit}}(D^t)$ *is* $\{\ \langle x, \mathsf{posit}, \langle\Phi, \phi\rangle\rangle \mid \langle \Phi, \phi\rangle$ *is an argument* $\}$ *and* $P_x^{\mathsf{concede}}(D^t)$ *is* $\{\ \langle x, \mathsf{concede}, \phi\rangle \mid \phi \notin \Sigma^t\ \}$.

Note that it is straightforward to check conformance with the protocol as the protocol only refers to public elements of the dialogue (i.e. it does not refer to perbases). For instance, the dialogue in Ex. 2 conforms to the protocol.

In general, a **strategy** for agent x is a function $\mathsf{Strategy}_x : \mathcal{D} \mapsto \wp(\mathcal{M})$ that takes the dialogue D^t and returns a subset of the legal moves. A strategy is personal to an agent and the moves that it returns depends on the agent's private beliefs (i.e. its perbase Δ_x).

A well-formed dialogue is a dialogue that does not continue after terminated and that is generated by the strategy.

Definition 8. *A dialogue D^t is a* **well-formed dialogue** *iff, for all s ($s < t$), (1) D^s does not terminate at s and (2) if $\mathsf{Sender}(m_s) = \overline{x}$, then $m_{s+1} \in \mathsf{Strategy}_x(D^s)$.*

In the next section, we give a specific strategy for using inconsistent ontologies, and then discuss alternatives to it.

4 An Example of a Strategy

We will shortly give a specific strategy function using the following subsidiary notions.

Definition 9. *Let Ψ be a set of formulae. The set of arguments that can be formed from Ψ is $\mathsf{Args}(\Psi) = \{\langle \Phi, \psi \rangle \mid \Phi \subseteq \Psi$ and $\langle \Phi, \psi \rangle$ is an argument $\}$.*

The posit moves that occur after and including the last posit move for the topic are called live moves.

Definition 10. *For D^t, the set of live moves, $\mathsf{Live}(D^t)$, is*

$$\begin{array}{l} \{m_k \mid \text{ there is an } i \text{ s.t. } i \leq k \leq t \\ \quad\quad \text{and } m_i = \langle x, \mathsf{posit}, \langle \Phi, \mathsf{Topic}(D^t) \rangle \rangle \\ \quad\quad \text{and there is not a } j \text{ s.t. } (i < j \leq t \\ \quad\quad\quad\quad \text{and } m_j = \langle x, \mathsf{posit}, \langle \Psi, \mathsf{Topic}(D^t) \rangle \rangle)\} \end{array}$$

For instance, for Ex. 2, as sequence $m_1, m_2, ..$ is made, the move m_3 is live until the move m_6 is made.

Given a dialogue, we define as follows whether an argument can be a novel undercut to extend the dialogue, as in Ex. 2, where each undercut is novel when posited.

Definition 11. *$\langle \Phi, \phi \rangle$ is a* **novel undercut** *for D^t iff there is a set of posit moves $\{n_1, ..., n_k\} \subseteq \mathsf{Live}(D^t)$ s.t. (1) n_1 posits an argument for the topic, (2) $\langle \Phi, \phi \rangle$ is an undercut for the posit of n_k, (3) for each i, $(1 < i \leq k)$, n_i is a posit for an undercut for the posit of n_{i-1}, (4) for each i, $(1 \leq i \leq k)$, Φ is not a subset of the support of the posit of n_i.*

We break a dialogue into phases. Intuitively, a phase is started by an agent positing an argument for the topic, and ended either by the dialogue ending, or by the next move being another posit move for the topic. So the live moves are in the latest phase in the dialogue.

Definition 12. *Let γ be the topic of D^t. A sequence of moves $m_i, ..., m_k$ is a* **phase** *in D^t iff $m_i, ..., m_k$ is a subsequence of D^t (i.e. D^t is $m_1, ..., m_i, ..., m_k, ..., m_t$ where $1 \leq i \leq k \leq t$), and m_i is a posit for γ and either t is k or m_{k+1} is a posit of γ and for all j s.t. $i < j \leq k$, m_j is not a posit for γ.*

$$\mathsf{Strategy}_x(D^t) = \begin{cases} S_x^{\mathsf{counter}}(D^t) & \text{iff } S_x^{\mathsf{counter}}(D^t) \neq \emptyset \\ S_x^{\mathsf{arg}}(D^t) & \text{iff } S_x^{\mathsf{arg}}(D^t) \neq \emptyset \text{ and } S_x^{\mathsf{counter}}(D^t) = \emptyset \\ S_x^{\mathsf{concede}}(D^t) & \text{iff } S_x^{\mathsf{concede}}(D^t) \neq \emptyset \\ & \quad\text{and } S_x^{\mathsf{counter}}(D^t) = \emptyset \text{ and } S_x^{\mathsf{arg}}(D^t) = \emptyset \\ \langle x, \mathsf{close}, \mathsf{Topic}(D^t)\rangle & \text{otherwise} \end{cases}$$

Fig. 1. The strategy function selects moves according to the following preference ordering (starting with the most preferred): posit moves, concede moves, and close moves. The conditions on the r.h.s. of each iff statement above imposes this ordering.

To ensure that each concede move is relevant, the formula being conceded must have an atom in common with a formula already in the commitment store or with the topic of the dialogue in order to ensure that it can potentially be used with other formulae in a posit move.

Definition 13. *A formula ϕ is* **relevant** *in D^t iff* $\mathsf{Names}(\phi) \cap \mathsf{Names}(\mathsf{Topic}(D^t)) \neq \emptyset$ *or* $\exists \psi \in \Sigma^t$ *s.t.* $\mathsf{Names}(\phi) \cap \mathsf{Names}(\psi) \neq \emptyset$.

We also require the following subsidiary functions. Essentially, S_x^{arg} gives the posits for the topic of the dialogue that have not been posited before, S_x^{counter} gives the undercuts for any argument in the current phase that have not been given so far in the current phase, and S_x^{concede} gives any formulae from its perbase or the lexbase that is relevant to the topic of the dialogue or to any formula already used in the dialogue.

Definition 14. *For the strategy function given in Fig. 1, we require the following sets of moves where $y \in \{1, 2\}$.*

$$\begin{aligned} S_x^{\mathsf{arg}}(D^t) = \{\langle x, \mathsf{posit}, \langle \Phi, \phi\rangle\rangle \in P_x^{\mathsf{posit}}(D^t) \mid \\ \langle \Phi, \phi\rangle \in \mathsf{Args}(\Delta_x \cup \Sigma^t) \\ \textit{and } \phi = \mathsf{Topic}(D^t) \\ \textit{and } \neg\exists m_i \textit{ s.t. } m_i \textit{ appears in } D^t \\ \textit{and } m_i = \langle y, \mathsf{posit}, \langle \Phi, \phi\rangle\rangle\} \end{aligned}$$

$$\begin{aligned} S_x^{\mathsf{counter}}(D^t) = \{\langle x, \mathsf{posit}, \langle \Phi, \phi\rangle\rangle \in P_x^{\mathsf{posit}}(D^t) \mid \\ \langle \Phi, \phi\rangle \in \mathsf{Args}(\Delta_x \cup \Sigma^t) \\ \textit{and } \langle \Phi, \phi\rangle \textit{ is a novel undercut for } D^t\} \end{aligned}$$

$$\begin{aligned} S_x^{\mathsf{concede}}(D^t) = \{\langle x, \mathsf{concede}, \phi\rangle \in P_x^{\mathsf{concede}}(D^t) \mid \\ (\phi \in \Delta_x \textit{ or } \phi \in \Pi) \\ \textit{and } \phi \textit{ is relevant in } D^t\} \end{aligned}$$

For the strategy defined in Fig. 1, a posit for the topic is made if possible. Then, the agents exhaustively present undercuts to this, and by recursion, undercuts to undercuts. When this is exhausted, the first phase has finished. If another posit for the topic can be made, then the second phase starts, and undercuts to this, and by recursion, undercuts to undercuts are exhaustively presented, thereby bringing the second phase to a close.

Subsequent phases are constructed accordingly. The dialogue in Ex. 2 is generated by this strategy.

Now, we consider how to evaluate these dialogues. The set of arguments posited in a phase is called a constellation.

Definition 15. *Let $m_i, ..., m_k$ be a phase in D^t. X is the* **constellation** *for $m_i, ..., m_k$ iff $X = \{\langle \Phi, \alpha \rangle \mid m_j = \langle x, \mathsf{posit}, \langle \Phi, \alpha \rangle \rangle$ and $m_j \in \{m_i, ..., m_k\}\}$.*

Intuitively, a dialogue supports the topic iff there is a constellation that can be obtained from a phase of the dialogue and that the constellation can be arranged as an argument tree with an undefeated root argument for the topic of the dialogue. For this property, we require the following: A **complete** argument tree is an argument tree (N, E, X, f) such that if there is a node $n \in N$, and an argument $A \in X$ where A undercuts $f(n)$ and there is not a node $n' \in N$ such that n' is on the branch from the root to n and the support of A is a subset of the support of $f(n')$, then there is a child m of n such that $f(m)$ is A.

Definition 16. *For a dialogue D^t where $\mathsf{Topic}(D^t) = \alpha$, D^t* **supports** *α iff there is a phase $m_i, ..., m_k$ in D^t s.t. X is the constellation for $m_i, ..., m_k$ and there is a complete argument tree for α (N, E, X, f) s.t. its root argument is undefeated.*

The dialogue in Ex. 2 has two phases and supports $C(a)$.

Example 3. $\Delta_1 = \{\neg D(a), D \sqsubseteq A, A \sqsubseteq \neg C\}$; $\Delta_2 = \{D(a), D \sqsubseteq C, D \sqsubseteq E, E \sqsubseteq \neg A\}$; $\Pi = \emptyset$.

$$\begin{array}{l}\langle 1, \mathsf{query}, C(a) \rangle \\ \langle 2, \mathsf{posit}, \langle \{D(a), D \sqsubseteq C\}, C(a) \rangle \rangle \\ \langle 1, \mathsf{posit}, \langle \{D \sqsubseteq A, A \sqsubseteq \neg C\}, D \sqsubseteq \neg C \rangle \rangle \\ \langle 2, \mathsf{posit}, \langle \{D \sqsubseteq E, E \sqsubseteq \neg A\}, D \sqsubseteq \neg A \rangle \rangle \\ \langle 1, \mathsf{posit}, \langle \{\neg D(a)\}, \neg D(a) \rangle \rangle\end{array}$$

Here there is just one phase $(m_2, ..., m_5)$. Agent 2 gives an argument, then Agent 1 gives an undercut to it, and then Agent 2 gives an undercut to that. Finally, Agent 1 comes back with an undercut to the first argument by Agent 2. As a result, $C(a)$ is not supported.

So each dialogue generated with this strategy ensures that all the possible arguments relevant to the query are presented, and these arguments can be assessed to determine whether the query is supported in the dialogue. Because the strategy returns choices of moves, the dialogue is not necessarily unique given a query, but the alternative dialogues that can be obtained given a query are isomorphic, and hence the constellations will be the same.

With the protocol, we can also define alternative strategies that ensure alternative useful behaviour. For instance, we could define a strategy that stops the dialogue when a phase has occurred that alone can be used to demonstrate support for the topic, or we could define a strategy that also gives the arguments for the negation of the topic, or we could define a strategy that only allows a concede move of a formula when that formula is consistent with its perbase.

5 Properties of Dialogical Argumentation

The constraints on the strategy function are such that we can show that all dialogues terminate (as agents' perbases are finite, hence there are only a finite number of different moves that can be generated and agents cannot repeat these moves *ad infinitum*).

Proposition 1. *For any well-formed dialogue D^t, there exists a D^u s.t. $t \leq u$ and D^u terminates at u and D^u extends D^t (i.e. the first t moves of D^u are given by D^t).*

A dialogue is sound if and only if, if an argument is generated by the dialogue, then it can also be constructed from the union of the perbases and the lexbase.

Definition 17. *Let D^t be a well-formed dialogue. We say that D^t is* **sound** *iff, for each s, if $s \leq t$ and $m_s = \langle x, \mathsf{posit}, \langle \Phi, \phi \rangle \rangle$, then $\langle \Phi, \phi \rangle$ is an argument s.t. $\Phi \subseteq (\Delta_x \cup \Delta_{\overline{x}} \cup \Pi)$.*

When an agent posits an argument, it must be able to construct the argument from its perbase and the commitment store. This is clear from the definition of the strategy. From this, and the fact that the commitment stores are only updated when a posit or concede move is made, we get that a commitment store is always a subset of the union of the perbases and the lexbase. From these observations, we get soundness.

Proposition 2. *If D^t is a well-formed dialogue, then D^t is sound.*

Similarly, a dialogue is complete if and only if, if the dialogue terminates at t and it is possible to construct an argument for the topic of the dialogue from the union of the perbases and lexbase, then that argument will eventually be posited by one of the agents.

Definition 18. *Let D^t be a well-formed dialogue and $\mathsf{Topic}(D^t) = \gamma$. We say that D^t is* **complete** *iff, if there is a argument $\langle \Phi, \gamma \rangle$ s.t. $\Phi \subseteq (\Delta_x \cup \Delta_{\overline{x}} \cup \Pi)$, then there is a move $\langle x, \mathsf{posit}, \langle \Phi, \gamma \rangle \rangle$ in D^t.*

In order to show that all dialogues are complete, we need some further lemmas. The first states: If an agent cannot produce, given their perbase and the commitment store, an argument for the topic of the dialogue, then the strategy forces them to concede formulae from their perbase and the lexbase, thus adding to the commitment store.

Lemma 1. *Let D^t be a well-formed dialogue with $\mathsf{Topic}(D^t) = \gamma$. If $S_x^{\mathsf{arg}}(D^t) = \emptyset$ and $S_x^{\mathsf{counter}}(D^t) = \emptyset$ and there is a $\beta \in \Delta_x \cup \Pi$ s.t. β is relevant for D^t and $\beta \notin \Sigma^t$ then $\langle x, \mathsf{concede}, \beta \rangle \in \mathsf{Strategy}_x(D^t)$.*

Following from the above lemma, we obtain the following lemma that says if there is an argument for the topic of the dialogue that can be obtained by pooling the agents' perbases and the lexbase, then, once the dialogue has terminated, there is the support for this argument in the union of the agent's perbase with the commitment store.

Lemma 2. *Let D^t be a well-formed dialogue that terminates at t with $\mathsf{Topic}(D^t) = \gamma$. If there is a $\Phi \subseteq (\Delta_x \cup \Delta_{\overline{x}} \cup \Pi)$ s.t. $\langle \Phi, \gamma \rangle$ is a argument, then $\Phi \subseteq (\Delta_x \cup \Sigma^t)$*

The next lemma says that if there is an argument for the topic of the dialogue that can be obtained from an agent's perbase and the commitment store, then the strategy will force the posit of that argument at some point in the dialogue.

Lemma 3. *Let D^t be a well-formed dialogue that terminates at t with $\mathsf{Topic}(D^t) = \gamma$. If there is a $\Phi \subseteq (\Delta_x \cup \Sigma^t)$ s.t. $\langle \Phi, \gamma \rangle$ is an argument, then there is an s s.t. $s < t$ and $m_s = \langle x, \mathsf{posit}, \langle \Phi, \gamma \rangle \rangle$.*

Using the above lemmas, it is straightforward to now show that dialogues are complete.

Proposition 3. *If D^t is a well-formed terminated dialogue, then D^t is complete.*

A dialogue is faithful if it supports the topic iff the arguments that can be constructed from the union of the perbases and the lexbase can be arranged as a complete argument tree for the topic where the root argument is undefeated.

Definition 19. *Let D^t be a well-formed dialogue, $\mathsf{Topic}(D^t) = \gamma$, and $X = \mathsf{Args}(\Delta_x \cup \Delta_{\overline{x}} \cup \Pi)$. We say that D^t is* **faithful** *when the following equivalence holds.*

D^t supports γ iff
there is a complete argument tree (N, E, X, f) for γ
where $f(n)$ is undefeated for root n

From completeness and soundness, for topic γ, we get that $\langle \Phi, \gamma \rangle$ is an argument from the union of the agents' perbases and the lexbase iff there is a posit move of $\langle \Phi, \gamma \rangle$ in the dialogue. Furthermore, there is exactly one phase for each of these arguments $\langle \Phi, \gamma \rangle$. We can then generalize the completeness and soundness results so that for each phase, if $\langle \Phi, \gamma \rangle$ is the argument that starts the phase, then the posit moves made in the phase contain exactly, the undercuts of $\langle \Phi, \gamma \rangle$, and by recursion, the novel undercuts to each undercut, that could be obtained from the union of the agents' perbases and the lexbase. Therefore, each phase is isomorphic to a complete argument tree for the topic that can be obtained from the union of the agents' perbases and the lexbase, and each complete argument tree for the topic that can be obtained from the union of the agents' perbases and the lexbase is isomorphic to a phase. Hence, we get the following.

Proposition 4. *If D^t is a well-formed terminated dialogue, then D^t is faithful.*

A corollary of this proposition is that all the minimal inconsistent subsets of the union of the agents' ontologies that involve the topic of the dialogue can be recovered from the commitment store of the dialogue. In other words, from each phase of the dialogue, the minimal inconsistent subsets involving the query can be obtained from the posit starting the phase, and the undercuts to this posit.

So in this section, we have shown that the dialogues always terminate, they are sound (any argument posited is an argument that can come from the union of the agents' ontologies plus the lexbase), they are complete (any argument for the topic obtainable from the union of the agents' ontologies plus the lexbase, is posited in the dialogue), they are faithful (any argument for the topic shown to be undefeated given the union of the agents' ontologies plus the lexbase, is shown to be undefeated in a phase in the dialogue, and *vice versa*). These properties mean that the dialogical argumentation is equivalent to argumentation with the union of the agents' ontologies plus the lexbase, but with the advantage that it is not necessary to copy all of each ontology to each agent in order to undertake the argumentation. Rather, just enough knowledge is exchanged in the posit and concede moves for the argument trees to be implicitly constructed in the dialogue.

6 Conclusions

We have presented a dialogical argumentation framework for using multiple ontologies. In comparison with the maxcon approach, we do not lose information, rather we keep it all, and we do not make arbitrary choices. Furthermore, any inference that can be obtained from the maxcon approach can be obtained from our approach, but not vice versa. In comparison with the oracle approach, we may get inferior inferences (i.e. inferences that with the benefit of some oracle are not deemed to be good), but the significant advantage here is that we do not need to copy and merge all of the ontology for each agent to use knowledge from other agents' ontologies.

There are other proposals for argumentation with ontologies. In [10, 13], all the ontological knowledge is in a centralized location, and so they do not get the advantages that come from using dialogues, and in [12], dialogues are used for discussing ontology alignments, but not for querying the ontological knowledge.

Another advantage of our approach is that it allows an agent to determine how its perbase differs from another. This can then be used by the agent to decide how to update its perbase. For instance, if it regards the other agent as more reliable, or if it has had the same conflict with a number of agents, it may choose to delete some of its own knowledge.

Our system also allows the definition of alternative strategies that ensure alternative intelligent behaviour. For instance, we can define a more efficient strategy that only builds a pruned version of the argument tree and yet still produces faithful dialogues. Also we can refine how agents chose to concede a move from the lexbase (e.g. by only allowing a formula to be conceded if it is consistent with the agent's perbase, or by more tightly coupling concession to the search for premises for arguments and counterarguments).

Our proposal is influenced by [5], but it does involve some substantial developments over it: (1) That paper was for a simple propositional defeasible logic whereas this paper is for much richer description logics; (2) That paper was only about finding arguments whereas this paper is about the more complex issue of finding warranted arguments; (3) That paper has different moves, protocol and strategy to this paper; (4) This paper has a more general framework based on phases that is valuable for supporting and auditing diverse protocols and strategies for arguing about ontologies.

The dialogues generated by our system allow agents to jointly construct arguments for a topic and to determine if this topic is supported given the union of available knowledge. As far as we are aware, there are only three other dialogue systems that share this same aim and have been shown to have similar properties to ours (i.e. are faithful) [3, 16, 6]; However, none deal with ontological knowledge and the first two impose restictions on the distribution of formulae between the agents. [14] also consider completeness properties for general classes of protocol that allow agents to jointly construct argument trees, but they do not allow the joint construction of arguments nor do they consider ontological knowledge.

In future work, we will develop a range of more refined strategies including for conceding formulae from the lexbase. We will also extend our system to address wider issues concerning semantic heterogenity arising between ontologies.

References

1. L Amgoud and C Cayrol. A model of reasoning based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–216, 2002.
2. F Baader, D L McGuiness, D Nardi, and P Patel-Schneider, editors. *Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2002.
3. J Bentahar, R Alam, and Z Maamar. An argumentation-based protocol for conflict resolution. In *KR2008-workshop on Knowledge Representation for Agents and Multi-Agent Systems*, pages 19–35, 2008.
4. P Besnard and A Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.
5. E Black and A Hunter. A generative inquiry dialogue system. In *6th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1010–1017, 2007.
6. E Black and A Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19(2):173–209, 2009.
7. P Dung, R Kowalski, and F Toni. Dialectical proof procedures for assumption-based admissible argumentation. *Artificial Intelligence*, 170:114–159, 2006.
8. G Flouris, Z Huang, J Z Pan, D Plexousakis, and H Wache. Inconsistencies, negations and changes in ontologies. In *21st AAAI Conf.*, pages 1295–1300, 2006.
9. A García and G Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Prog.*, 4(1):95–138, 2004.
10. S Gómez, C Chesñevar, and G Simari. An argumentative approach to reasoning with inconsistent ontologies. In *Knowledge Representation and Ontologies Workshop*, pages 11–20, 2008.
11. Z Huang, F van Harmelen, and A ten Teije. Reasoning with inconsistent ontologies. In *19th Int. Joint Conf. on Artificial Intelligence*, pages 454–459, 2005.
12. L Laera, I Blacoe, V Tamma, T Payne, J Euzenat, and T Bench-Capon. Argumentation over ontology correspondences in MAS. In *6th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1293–1300, 2007.
13. M Moguillansky, N Rotstein, and M Falappa. A theoretical model to handle ontology debugging & change through argumentation. In *2nd Int. Workshop on Ontology Dynamics*, 2008.
14. S Parsons, P McBurney, E Sklar, and M Wooldridge. On the relevance of utterances in formal inter-agent dialogues. In *6th Int. Joint Conf. on Autonomous Agents and Mutli-Agent Systems*, pages 1002–1009, 2007.
15. H Prakken and G Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logic*, 7:25–75, 1997.
16. M Thimm and G Kern-Isberner. A distributed argumentation framework using defeasible logic programming. In *2nd Int. Conf. on Computational Models of Argument*, pages 381–392, 2008.