KR2008-Workshop on Knowledge
Representation for Agents and Multi-Agent Systems, pp. 19-35

An Argumentation-based Protocol for Conflict Resolution

Jamal Bentahar[1], Rafiul Alam[1], and Zakaria Maamar[2]

[1] Concordia University, Concordia Institute for Information Systems Engineering, Canada
[2] Zayed University, College of Information Systems, UAE

Abstract. This paper proposes an argumentation-based protocol for resolving conflicts between agents. These agents use assumption-based argumentation in which arguments are built from a set of rules and assumptions using backward deduction. Beyond arguments agents can handle, we propose the notion of *partial arguments* along with *partial attack* and *partial acceptability*. These notions allow agents to reason about partial information. Unlike existing protocols, this protocol merges inquiry and persuasion stages. In addition, by building and reasoning about partial arguments, agents can jointly find arguments supporting a new solution for their conflict, which is not known by any of them individually. Furthermore, during persuasion, agents can acquire new beliefs and drop attacked ones. The protocol is formally specified as a set of simple dialogue rules about which agents can reason using argumentation. We also provide termination, soundness and completeness results of the proposed protocol.

1 Introduction

In recent years, argumentation has been used in many applications such as legal reasoning, automatic negotiation, persuasion, online debates, medical applications, and Web services [19]. In multi-agent systems, some interesting argumentation-based protocols for persuasion and inquiry have been proposed. [2] proposes the Persuasive Argument for Multiple Agents (PARMA) Protocol, which enables participants to propose, attack, and defend an action or course of actions. This protocol is specified using logical consequence and denotational semantics, which maps statements in the syntax to mathematical entities. The focus of this work is more about the semantics of the protocol rather than the dynamics of interactions. [4] proposes a dialogue-game inquiry protocol that allows two agents to share knowledge in order to construct an argument for a specific claim. The protocol is declaratively specified as a function that returns the set of legal moves. A strategy function is also specified to allow an agent to select exactly one of the legal moves to make. This protocol considers only pure inquiry where no conflicting goals are identified. In the context of agent communication, [17] proposes an alternative view on argumentation and persuasion using a coherence theory. Argument generation, evaluation and integration within this theory are discussed in a pragmatic way. However, no protocol about using the proposed framework has been specified. [6] develops a dynamic, situation calculus-based argumentation model in which protocols describe, in a declarative way, which speech acts are legal in a particular state. The model is used to analyze a formal disputation. Except the [4]'s protocol, the others do not consider agents' strategies and the correctness and completeness properties are not discussed.

In a series of papers, researchers from Toulouse and Liverpool have developed an approach to specify persuasion and inquiry protocols. Particularly [16] uses propositional logic to define the underlying argumentation framework and only three locutions have been used in these protocols: *Assert*, *Accept*, and *Challenge*. The purpose is mainly to discuss the *pre-determinism* issue (i.e. to what extent the outcomes of dialogues are *predetermined* when using these protocols). The idea is to check if these outcomes are determined by agents' knowledge and the order agents utter locutions. The authors show that these protocols are not complete in the sense that the generated dialogues are not pre-determined. Also, agents' strategies on how to use the protocols are not considered, and persuasion and inquiry are dealt with as two different protocols without connection between them. [18] proposes formal dialogue games for persuasion protocols where arguments are assumed to be trees of deductive and/or defeasible inferences. Each dialogue move either attacks or surrenders to some earlier move of the addressee. The motivation of the framework is to ensure coherence and flexibility of dialogues. The protocol notion is specified as a function defining the legal moves, and the framework does not consider the agents' strategies.

All the protocols defined in the aforementioned proposals are either pure persuasion or pure inquiry. Also, the proposed persuasion protocols are not complete in the sense of pre-determinism. Except a few proposals such as [4], the notion of agents' strategies on how to use these protocols is disregarded. The purpose of this paper is to address these limitations. The contribution of this work is the proposition of a new sound and complete protocol combining persuasion and inquiry for conflict resolution. Agents use assumption-based argumentation in which arguments are built from a set of rules and assumptions using backward deduction. This protocol is different from all the aforementioned proposals in many points. It is operationally specified as simple *if-then rules* with conditions whose values determine the reply an agent can perform. The agent strategy is used when evaluating these conditions based on private agents' beliefs and publicly exchanged information between communicating agents. The whole protocol can be built by simply combining these rules. In addition, there are two original ideas behind it: (1) the notion of *partial arguments and their "acceptability" statuses* allowing agents to reason about incomplete information; and (2) the combination of persuasion and inquiry allowing agents to jointly find out new solutions for their conflicts that cannot be found by any of them individually. Consequently, new solutions can emerge by exchanging arguments and partial arguments, which allows resolving the problem of pre-determinism.

Section 2 presents the argumentation model and the notions of partial arguments and conflicts. Section 3 presents the protocol specification along with the different dialogue rules and strategies, and discusses its properties. Section 4 gives an illustrative example of the protocol. Section 5 concludes the paper by discussing some related work.

2 Argumentation Model

2.1 Language and Background

This section discusses the key elements of the formal argumentation system. Many argumentation systems have been proposed in the literature such as the abstract argumentation [10], logic-based argumentation [1, 3, 16], preference value-based argumentation [13], logic-programming-based argumentation [14, 12], and facts and rules-based argumentation [4] (see [7] for a survey). In this paper, we use assumption-based argumentation adapted from [5] and [8]. Assumption-based argumentation has been proven to be a powerful mechanism to understand commonalities and differences amongst many existing frameworks for non-monotonic reasoning, including logic programming [5]. This framework is built upon Dung's abstract argumentation by instantiating the primitive notions of *argument* and *attack* using notions of *deductive system* and corresponding *deductions*, *assumptions and contrary of assumptions*. Mechanisms for computing Dung's argumentation semantics [10] have been developed for this framework with some computational advantages, particularly in terms of avoiding re-computation by filtering out assumptions that have already been defended or defeated [9].

We use a formal language $\mathcal{L}$ to express agents' beliefs. This language consists of countably many sentences (or wffs). Also, the language is associated with an abstract *contrary mapping* like the one used in [11] (the negation is an example of this mapping, so the contrary of p is $\neg p$). We do not assume this mapping to be necessarily symmetric. $\bar{x}$ denotes an arbitrary contrary of a wff x. By arbitrary we mean we do not need to specify this contrary. Agents build arguments using their beliefs. The set $Arg(\mathcal{L})$ contains all those arguments.

Definition 1 (Argumentation Framework). *An assumption-based argumentation framework is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ where:*

- *$(\mathcal{L}, \mathcal{R})$ is a deductive system, with a countable set $\mathcal{R}$ of inference rules,*
- *$\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set, whose elements are referred to as assumptions,*
- *$^-$ is a total mapping from $\mathcal{A}$ into $2^{\mathcal{L}} - \emptyset$, where $^-\alpha$ is the non-empty set of contraries of α and $\bar{\alpha}$ is an arbitrary contrary of α ($\bar{\alpha} \in^- \alpha$).*

We will assume that the inference rules in $\mathcal{R}$ have the form: $c_0 \leftarrow c_1, \ldots, c_n$ with $n > 0$ or the form c_0 where each $c_i \in \mathcal{L}$ $(i = 0, \ldots, n)$.
c_0 is referred to as the head and $c_1, \ldots, c_n$ as the body of a rule $c_0 \leftarrow c_1, \ldots, c_n$. The body of a rule c_0 is considered to be empty. We will restrict attention to *flat assumption-based frameworks*, such that if $c \in \mathcal{A}$, then there exists no inference rule of the form $c \leftarrow c_1, \ldots, c_n \in \mathcal{R}$. Before defining the notions of argument and attack relation, we give here a formal definition of the backward deduction that is used in this framework.

The backward deduction can be represented by a top-down proof tree linking the conclusion to the assumptions. The root of the tree is labelled by the conclusion and the terminal nodes are labelled by the assumptions. For every non-terminal node in the tree, there is an inference rule whose head matches the sentence labelling the node. The children of the node are labelled by the body of the inference rule. Consequently, the backward deduction can be represented as a set of steps $S_1, \ldots, S_m$ and in each step we have a set of sentences to which we can apply inference rules because each sentence matches the head of a rule. From each step to the next one, the procedure consists of selecting one sentence and replacing it, if the sentence is not an assumption, by the body of the corresponding inference rule. The selection strategy is represented by the following function:

$$SS : Step \rightarrow \mathcal{L}$$

where $Step = \{S_1, \dots, S_m\}$

Definition 2 (Backward Deduction). *Given a deduction system $(\mathcal{L}, \mathcal{R})$ and a selection strategy function SS, a backward deduction of a conclusion c from a set of assumptions X is a finite sequence of sets $S_1, \dots, S_m$, where $S_1 = \{c\},\ S_m = X$, and for every $1 \leq i < m$:*
***1.** If $SS(S_i) \notin X$ then $S_{i+1} = S_i - \{SS(S_i)\} \cup B$ for some inference rules of the form $SS(S_i) \leftarrow B$.*
***2.** Otherwise, $S_{i+1} = S_i$.*

Definition 3 (Argument). *Let $X \subseteq \mathcal{A}$ be a consistent subset of assumptions (i.e. X does not include a formula and one of its contraries), and let c be a sentence in $\mathcal{L}$. An argument in favor of c is a pair (X, c) such that c is obtained by the backward deduction from X. c is called the conclusion of the argument.*

In the rest of the paper, arguments will be denoted as pairs (X, c) when the set of premises is needed for our analysis. Otherwise, arguments will be simply denoted by $a, b, d, \dots$. The notion of conflicts between arguments is captured via the attack relation. The attack relation we define here is more general than the one used in the original assumption-based argumentation in the sense that we allow attacking, not only the assumptions, but also the conclusion of the argument.

Definition 4 (Attack Relation). *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments over the argumentation framework. The attack relation between arguments $\mathcal{AT} \subseteq Ar \times Ar$ is a binary relation over Ar that is not necessarily symmetric. An argument (X, c) attacks another argument (X', c') denoted by $\mathcal{AT}((X, c), (X', c'))$ iff c is a contrary of c' or c is a contrary of a sentence $c'' \in X'$.*

In our protocol (Section 3), by using this attack relation, agents can try to win the dispute by trying different arguments for the same conclusion. For example, if an assumption is attacked, and cannot be defended, the agent can try to defend the conclusion using another assumption.

As conflicts between arguments might occur, we need to define what an acceptable argument is. Different semantics for argument acceptability have been proposed in [10]. These are stated in definitions 5, 6, 7, and 8.

Definition 5 (Defense). *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments over the argumentation framework, and let $S \subseteq Ar$. An argument a is defended by S iff $\forall\ b \in Ar$ if $\mathcal{AT}(b, a)$, then $\exists\ c \in S : \mathcal{AT}(c, b)$.*

Definition 6 (Admissible Set). *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments over the argumentation framework. A set $S \subseteq Ar$ of arguments is admissible iff:*
1) $\nexists\ a, b \in S$ such that $\mathcal{AT}(a, b)$ and
2) $\forall a \in S$ a is defended by S.

In other words, a *set of arguments* is admissible iff it is conflict-free and can counter-attack every attack.

Definition 7 (Characteristic Function). *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments and let S be an admissible set of arguments. The characteristic function of the argumentation framework is:*

$$F : 2^{Ar} \rightarrow 2^{Ar}$$
$$F(S) = \{a \mid a\ is\ defended\ by\ S\}$$

Definition 8 (Acceptability Semantics). *Let S be an admissible set of arguments, and let F be the characteristic function of the argumentation framework.*

– *S is a complete extension iff $S = F(S)$.*
– *S is the grounded extension iff S is the minimal (w.r.t. set-inclusion) complete extension (the grounded extension corresponds to the least fixed point of F).*
– *S is a preferred extension iff S is a maximal (w.r.t. set-inclusion) complete extension.*

Now we can define what are the acceptable arguments in our system.

Definition 9 (Acceptable Arguments). *Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments, and let G be the grounded extension in the argumentation framework. An argument a over Ar is acceptable iff $a \in G$.*

According to this acceptability semantics, which is based on the grounded extension, if we have two arguments a and b such that $\mathcal{AT}(a, b)$ and $\mathcal{AT}(b, a)$, then a and b are both non-acceptable. This notion is important in persuasion dialogues since agents should agree on an acceptable opinion, which is supported by an acceptable argument when a conflict arises. However, during the argumentative conversation, agents could use non-acceptable arguments as an attempt to change the status of some arguments previously uttered by the addressee, from acceptable to non-acceptable. This idea of using non-acceptable arguments in the dispute does not exist in the persuasion and inquiry protocols in the literature. For this reason, we introduce two new types of arguments based on the preferred extensions to capture this notion. We call these arguments *semi-acceptable and preferred semi-acceptable arguments*.

Definition 10 ((Preferred) Semi-Acceptable Arguments). *Let G be the grounded extension in the argumentation framework, and let $E_1, \ldots, E_n$ be the preferred extensions in the same framework. An argument a is:*

– *Semi-acceptable iff $a \notin G$ and $\exists E_i, E_j$ with $(1 \leq i, j \leq n)$ such that $a \in E_i \wedge a \notin E_j$.*
– *Preferred semi-acceptable iff $a \notin G$ and $\forall E_i$ $(1 \leq i \leq n)$ $a \in E_i$.*

In other words, an argument is *semi-acceptable* iff it is not acceptable and belongs to some preferred extensions, but not to all of them. An argument is *preferred semi-acceptable* iff it is not acceptable and belongs to all the preferred extensions. Preferred semi-acceptable arguments are stronger than semi-acceptable and grounded arguments are the strongest arguments in this classification.

Proposition 1. *the arguments defending themselves by only themselves against all the attackers (the set of attackers is supposed to be non-empty) are semi-acceptable.*

Definition 11 (Eliminated Arguments). *An argument is eliminated iff it does not belong to any preferred extension in the argumentation framework.*

We can easily prove that an argument is *eliminated* iff it is not acceptable, not preferred semi-acceptable, and also not semi-acceptable. Consequently, arguments take four exclusive statuses namely acceptable, preferred semi-acceptable, semi-acceptable, and eliminated. The dynamic nature of agents interactions is reflected by the changes in the statuses of *uttered arguments*. Let us now define the notions of conflict and conflict resolution.

Definition 12 (Conflicts). *Let α and β be two argumentative agents and AF_α and AF_β their respective argumentation frameworks. These two frameworks share the same contrary relation and the same rules, but not necessarily the same assumptions. There is a conflict between α and β iff one of them (e.g., α) has an acceptable argument (X, c) relative to AF_α and the other (i.e., β) has an acceptable argument $(X', \bar{c})$ relative to AF_β. We denote this conflict by $\alpha_c \ncong \beta_{\bar{c}}$ ($\bar{c} \in^- c$).*

For example, in an e-business setting if c and $\bar{c}$ represent each a security policy s_1 and s_2 such that s_1 and s_2 cannot be used together, then there is a conflict if one agent has an acceptable argument for using s_1 while the other agent has an acceptable argument for using s_2. This conflict arises when both agents need to agree on which security policy to use. For simplification reasons and when the set of assumptions is not needed for our analysis, an argument a supporting a conclusion c will be denoted $a \uparrow c$.

Definition 13 (Conflicts Resolution). *A conflict between two agents is resolved after interaction iff they agree on a solution which is supported by an acceptable argument for both agents.*

In the aforementioned security example, the conflict is resolved iff (i) after interaction, one of the agents can build an acceptable argument from its knowledge base and the arguments exchanged during this interaction, supporting the use of the other policy, or (ii) when both agents agree on the use of a new policy such that each agent can build an acceptable argument, from its knowledge base and the exchanged arguments, supporting the use of this policy. The idea here is that by exchanging arguments, new solutions (and arguments supporting these solutions) can emerge. In this case, agents should update their beliefs by withdrawing attacked (i.e. eliminated) assumptions. However, there is still a possibility that each agent keeps its viewpoint at the end of the conversation.

2.2 Partial Arguments

The outcome of an interaction aiming to resolve a conflict depends on the status of the formula representing the conflict topic. As for arguments, a wff has four statuses depending on the statuses of the arguments supporting it (an argument supports a formula if this formula is the conclusion of that argument). A wff is *acceptable* if there exists an acceptable argument supporting it. If not, and if there exists a preferred semi-acceptable argument supporting it, then the formula is preferred semi-acceptable. Otherwise, the formula is semi-acceptable if a semi-acceptable argument supporting it exists, or eliminated if such an argument does not exist. Let St be the set of these statuses. We define the following function that returns the status of a wff with respect to a set of arguments:

$$\Delta : \mathcal{L} \times 2^{Ar} \rightarrow St$$

To resolve conflicts, it happens that agents do not have complete information on some facts. In similar situations, they can build *partial arguments* for some conclusions out of their beliefs. We define a partial argument as follows:

Definition 14 (Partial Arguments). *Let $X \subseteq \mathcal{A}$ be a consistent subset of assumptions, and c a sentence in $\mathcal{L}$. A partial argument in favor of c is a pair denoted by $(X, c)_\partial$ such that $\exists Y \subseteq \mathcal{A}$ $(Y \neq \emptyset)$ and $(X \cup Y, c)$ is an argument.*

Example 1. $\mathcal{L} = \{p, q, r, t, m\}$, $\mathcal{R} = \{p \leftarrow q, r; p \leftarrow q, t, m\}$, $\mathcal{A} = \{q, r, t, m\}$. $(\{q\}, p)_\partial$ is a partial argument.

Because agents can use different sets of rules to build the same argument, it is clear that there are may be several sets of assumptions leading to this argument. Consequently, the set Y in Definition 14 is not unique. Agents can identify a set Y by identifying possible rules leading to the conclusion. When we need to refer to a given set Y (which is missing to complete the argument), we use the notation $(X, c)_\partial^Y$.

Example 2. The partial argument presented in example 1 can be completed by the assumption r or by the assumptions t and m. They can be denoted respectively by: $(\{q\}, p)_\partial^{\{r\}}$ and $(\{q\}, p)_\partial^{\{t,m\}}$

The idea behind building partial arguments is that in some situations, agents have partial information to build arguments for some conclusion, and the missing information can be obtained through interactions. If an agent misses a set of assumptions, it can check if the other agent can provide the missing part or a part of this missing part so that the complete argument could be jointly built progressively. This issue, already identified in [4], is a part of the inquiry dialogue and will be made clear in the protocol we define in the next section.

In the security scenario presented above, an example where partial arguments are needed is when the agent defending the security policy s_1 knows that the security policy s_2 that the other agent uses can be substituted by policy s_1 if some conditions are met when deploying s_2. Thus, this agent can build a partial argument supporting the fact that the second agent can use s_1. To be an argument, this partial argument needs the assumptions that implementing s_2 by the second agent meets these conditions.

As for arguments, we need to define the *status of partial arguments*. Our idea here, which is different from what is proposed in [4], is that if, considering the information an agent has at the current moment, there is no chance for the partial argument to be acceptable or at least to change the status of already uttered arguments, then there is no need to try to build such a partial argument. When the internal structure of these partial arguments is not needed, we use the notations a_∂ and a_∂^Y to denote a partial argument and a partial argument that can be completed by the set Y of assumptions respectively. The argument obtained by adding the assumptions Y to the partial argument a_∂^Y is denoted by $a_\partial^Y.Y$. The following definitions establish the status of partial arguments.

Definition 15 (Partial Attack). -

- $\mathcal{AT}(a_\partial^Y, b)$ *iff* $\mathcal{AT}(a_\partial^Y.Y, b)$
- $\mathcal{AT}(b, a_\partial^Y)$ *iff* $\mathcal{AT}(b, a_\partial^Y.Y)$
- $\mathcal{AT}(a_\partial^Y, b_\partial^{Y'})$ *iff* $\mathcal{AT}(a_\partial^Y.Y, b_\partial^{Y'}.Y')$

In words, a partial argument a_∂^Y attacks (is attacked by) an argument b iff $a_\partial^Y.Y$ attacks (is attacked by) b. Also, a partial argument a_∂^Y attacks another partial argument $b_\partial^{Y'}$ iff $a_\partial^Y.Y$ attacks $b_\partial^{Y'}.Y'$.

Definition 16 (Status of Partial Arguments). *A partial argument a_∂^Y is acceptable ((preferred) semi-acceptable, eliminated) iff $a_\partial^Y.Y$ is acceptable ((preferred) semi-acceptable, eliminated).*

Example 3. $\mathcal{L} = \{p, q, r, \bar{r}, s, t, u, \}$, $\mathcal{R} = \{p \leftarrow q; s \leftarrow r; t \leftarrow s; \bar{r} \leftarrow u\}$, $\mathcal{A} = \{q, r, u\}$.
The partial argument $(\emptyset, p)_\partial^{\{q\}}$ is acceptable. However the partial argument $(\emptyset, t)_\partial^{\{r\}}$ is not acceptable since the argument $(\{u\}, \bar{r})$ attacks the argument $(\{r\}, t)$.

Agents should consider the status of their partial arguments before using them. For example, if an agent has a partial argument a_∂^Y and by supposing that the assumptions in the set Y are true, the resulting argument $a_\partial^Y.Y$ is already eliminated by considering the arguments already uttered, there is no need to try to establish these assumptions. However, if this is not the case, the agent will ask the addressee about these assumptions. Also, if an agent has an acceptable partial argument for (resp. against) a conclusion, and an acceptable argument against (resp. for) this conclusion, this agent will utter its acceptable argument only if the acceptable partial argument cannot emerge from the interaction. The motivation behind this *general rule* is that if the partial argument is acceptable, then if it becomes a complete argument (by establishing the missing assumptions), the status of the existing acceptable argument will change to non-acceptable. So, the agent should consider first the partial acceptable argument.

3 Protocol for Resolving Conflicts

3.1 Agent Configuration

For simplification reason, we suppose that only two agents take part in the argumentative conversation to resolve their conflict. We denote participating agents by α and β. Agents share the same set of rules and each agent has a possibly inconsistent belief base $\mathcal{A}_\alpha$ and $\mathcal{A}_\beta$ respectively containing assumptions, where $\mathcal{A}_\alpha \cup \mathcal{A}_\beta = \mathcal{A}$ the set of assumptions in the argumentation framework.

Agents use their argumentation systems to decide about the next move to play (e.g., accept or attack the arguments uttered during their interactions). Agents strategies are based on these systems. When an agent accepts an argument that an addressee suggests, this agent updates its knowledge base by adding the elements of this argument and removing all the elements that attack this argument. Each agent α has also a commitment store (CS_α) publicly accessible for reading but only updated by the owner agent. Our protocol is to be used when a conflict is identified, for example as an outcome of a previous interaction. Consequently, when the protocol starts, the commitment stores of the two agents contain conflicting information. For example, $CS_\alpha = \{p\}$ and $CS_\beta = \{\bar{p}\}$ where p is a wff representing the conflict topic. Commitment stores are updated by adding arguments and partial arguments that the agents exchange. CS_α refers to the commitment store of agent α *at the current moment*.

The possibility for an agent α to build an acceptable argument a (respectively an acceptable partial argument a_∂^Y) from its knowledge base and the commitment store of the addressee β is denoted by $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a$ (respectively $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a_\partial^Y$). $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \not\rhd a$ (respectively $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \not\rhd a_\partial^Y$) means that agent α cannot build an acceptable argument a (respectively an acceptable partial argument a_∂^Y) from $\mathcal{A}_\alpha \cup CS_\beta$. For simplification reason, we associate the same symbols ($\trianglerighteq$ and $\ntrianglerighteq$) with (partial) preferred semi-acceptable and (partial) semi-acceptable arguments. However, agents consider first (partial) preferred semi-acceptable arguments.

3.2 Protocol Rules

In our framework, agents engage in persuasion and inquiry dialogues to resolve conflicts. We propose a persuasion-inquiry protocol, in which pre-determinism is considered. The protocol is modeled using a set of simple *dialogue rules* governing interactions between agents, in which each agent moves by performing utterances. These rules that correspond to dialogue games [15] are expressed as simple if-then rules that can be easily implemented. In this section, we define the notion of protocol and specify the protocol rules.

Definition 17 (Protocol). *A protocol is a pair $\langle \mathcal{C}, \mathcal{D} \rangle$ with $\mathcal{C}$ a finite set of allowed moves and $\mathcal{D}$ a set of dialogue rules.*

The moves in $\mathcal{C}$ are of n different types ($n > 0$). We denote by $M^i(\alpha, \beta, a, t)$ a move of type i played by agent α and addressed to agent β at time t regarding a content a. We consider four types of moves in our protocol: $Assert$, $Accept$, $Attack$, and $Question$. Generally, in the persuasion protocol agents exchange arguments. Except the $Question$ move whose content is not an argument, the content of other moves is an argument a ($a \in Arg(\mathcal{L})$). When replying to a $Question$ move, the content of $Assert$ move can also be a partial argument or "?" when the agent does not know the answer. We use another special move $Stop$ with no content. It could be played by an agent to stop the interaction.

Intuitively, a dialogue rule in $\mathcal{D}$ is a rule indicating the possible moves that an agent could play following a move done by an addressee. To make agents deterministic, we specify these rules using conditions that reflect the agents' strategies. Each condition C_j is associated with a single reply. This is specified formally as follows:

Definition 18 (Dialogue Rule). *A dialogue rule is either of the form:*

$$\bigwedge_{\substack{0<k\leq n_i \\ j\in J}} \left(M^i(\alpha, \beta, a, t) \wedge C_k \Rightarrow M_k^j(\beta, \alpha, a_k, t')\right)$$

where J is the set of move types, M^i and M^j are in $\mathcal{C}$ (M_k^j is the k^{th} move of type j), $t < t'$ and n_i is the number of allowed communicative acts that β could perform after receiving a move of type i from α; or of the form:

$$\bigwedge_{\substack{0<k\leq n \\ j\in J}} \big(C_k \Rightarrow M_k^j(\alpha, \beta, a_k, t_0)\big)$$

where t_0 is the initial time and n is the number of allowed moves that α could play initially.

In order to guarantee determinism and deadlock freedom, conditions C_j need to be mutually exclusive and exhaustive. Agents use their argumentation systems to evaluate, in a private manner, these conditions. These argumentation systems are based on the private agents' beliefs and public commitments recorded in the commitment stores.

To simplify the notations, we omit the time parameter from the moves and use the notation $\cup CS$ as an abbreviation of $CS_\alpha \cup CS_\beta$. Also, Ar denotes the set of all arguments that can be built by the two agents considering the union of there argumentation frameworks (the union of assumption sets) and the fact that they share the same rules. In our protocol, agents are not allowed to play the same move (with the same content) more than one time. We specify the different dialogue rules of our protocol as follows:

1- Initial Rule

$C_{in1} \Rightarrow Assert(\alpha, \beta, a)$

where:

$$\begin{aligned} C_{in1} = {} & \exists\, p, q \in \mathcal{L} : \\ & \alpha_p \ncong \beta_q \wedge \mathcal{AR}(\mathcal{A}_\alpha) \rhd a \wedge a \uparrow p \end{aligned}$$

The persuasion starts when a conflict is detected, for example as an outcome of a previous interaction. There is a conflict in the sense that agent α supports p and agent β supports q, a contrary of p. At first, one of the two agents asserts an acceptable argument supporting its position. In the remainder of this section, we suppose that the persuasion topic is represented by the wff p.

2- Assertion Rule

$$\begin{aligned} & Assert(\alpha, \beta, \mu) \wedge C_{as1} \Rightarrow Attack(\beta, \alpha, b) \quad \wedge \\ & Assert(\alpha, \beta, \nu) \wedge C_{as2} \Rightarrow Question(\beta, \alpha, Y) \wedge \\ & Assert(\alpha, \beta, \nu) \wedge C_{as3} \Rightarrow Accept(\beta, \alpha, a) \quad \wedge \\ & Assert(\alpha, \beta, \nu) \wedge C_{as4} \Rightarrow Stop(\beta, \alpha) \end{aligned}$$

where μ is an argument or partial argument, ν is an argument, partial argument, or "?" and:

$C_{as1} \;\; = Op_{as_1}^{at_1} \vee (\neg Op_{as_1}^{at_1} \wedge Op_{as_1}^{at_2})$

$$
\begin{aligned}
Op_{as_1}^{at_1} &= \exists\, b \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd b \\
&\quad \wedge\; \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b\}) \\
Op_{as_1}^{at_2} &= \exists\, b \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b \\
&\quad \wedge\; \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b\})
\end{aligned}
$$

$$
\begin{aligned}
C_{as2} &= \neg C_{as1} \wedge (Op_{as_2}^{qu_1} \vee (\neg Op_{as_2}^{qu_1} \wedge Op_{as_2}^{qu_2})) \\
Op_{as_2}^{qu_1} &= \exists\, b_\partial^Y,\ b_\partial^Y.Y \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd b_\partial^Y \\
&\quad \wedge\; \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b_\partial^Y.Y\}) \\
Op_{as_2}^{qu_2} &= \exists\, b_\partial^Y,\ b_\partial^Y.Y \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b_\partial^Y \\
&\quad \wedge\; \Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b_\partial^Y.Y\})
\end{aligned}
$$

$$
\begin{aligned}
C_{as3} &= \exists\, a \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a \wedge a \uparrow p \\
&\quad \wedge \neg Op_{as_2}^{qu_1} \wedge \neg Op_{as_2}^{qu_2}
\end{aligned}
$$

$$
\begin{aligned}
C_{as4} &= \neg Op_{as_1}^{at_1} \wedge \neg Op_{as_2}^{qu_1} \wedge \neg Op_{as_2}^{qu_2} \wedge \neg C_{as3} \\
&\quad \wedge \forall\, b \in Ar,\ \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b \Rightarrow \\
&\qquad\qquad \Delta(p, \cup CS) = \Delta(p, \cup CS \cup \{b\})
\end{aligned}
$$

In this rule, the content of $Assert$ could be an argument, partial argument, or "?". Indeed agents can use this move to assert new arguments in the initial rule or to reply to a question in the question rule, which is a part of *inquiry* in our protocol. The move that agent β can play as a reply to the $Assert$ move depends on the content of this assertion. When α asserts an argument or a partial argument, CS_α gets changed by adding the advanced (partial) argument. Agent β can attack agent α if β can generate an acceptable argument from its knowledge base and the α's commitment store so that this argument will change the status of the persuasion topic. Consequently, in this protocol agents do not attack only the last uttered argument, but any uttered argument during the interaction, which is still acceptable or (preferred) semi-acceptable ($Op_{as_1}^{at_1}$). This makes the protocol more flexible and efficient (for example agents can try different arguments to attack a given argument). If such an acceptable argument cannot be generated, β will try to generate a (preferred) semi-acceptable argument changing the status of p ($Op_{as_1}^{at_2}$). The idea here is that if β cannot make α's arguments eliminated, it will try to make them semi-acceptable or at least preferred semi-acceptable. This is due to the following proposition whose proof is straightforward from the definition of (preferred) semi-acceptable arguments.

Proposition 2. *If β plays the $Attack$ move with a semi-acceptable argument, then the status of the persuasion topic changes from acceptable to preferred semi-acceptable or semi-acceptable.*

We notice that in the Assertion rule changing the status of p is a result of an attack relation:

Proposition 3. *In Assertion rule we have:* $\forall\, b \in Ar$,
$\Delta(p, \cup CS) \neq \Delta(p, \cup CS \cup \{b\}) \Rightarrow \exists\, a \in \cup CS : \mathcal{AT}(b, a)$.

If β cannot play the $Attack$ move, then before checking the acceptance of an α's argument, it checks if no acceptable and then no (preferred) semi-acceptable argument in the union of the knowledge bases can attack this argument (inquiry part). For that, if β can generate a partial argument changing the status of p, then it will question α about the missing assumptions ($Op^{qu_1}_{as_2}$ and $Op^{qu_2}_{as_2}$). This new feature provides a solution to the "pre-determinism" problem identified in [16]. If such a partial argument does not exist, and if β can generate an acceptable argument supporting p, then it plays the $Accept$ move (C_{as3}).

Proposition 4. *An agent plays the Accept move only if it cannot play the Attack move and cannot play the Question move.*

Proof. See Appendix

Agent β plays the $Stop$ move when it cannot accept an α's argument and cannot attack it. This happens when an agent has a (preferred) semi-acceptable argument for p and the other a (preferred) semi-acceptable argument against p, so the status of p in the union of the commitment stores will not change by advancing the β's argument (C_{as4}). Finally, we notice that if the content of $Assert$ move is "?", β cannot play the $Attack$ move. The reason is that such an $Assert$ is played after a question in the Question rule, and agents play $Question$ moves only if an attack is not possible. By simple logical calculus, we can prove the following proposition:

Proposition 5. *In the protocol, an agent plays the* $Stop$ *move iff it cannot play another move.*

3- Attack Rule

$$
\begin{array}{l}
Attack(\alpha,\beta,a) \wedge C_{at1} \Rightarrow Attack(\beta,\alpha,b) \quad\quad \wedge \\
Attack(\alpha,\beta,a) \wedge C_{at2} \Rightarrow Question(\beta,\alpha,Y) \wedge \\
Attack(\alpha,\beta,a) \wedge C_{at3} \Rightarrow Accept(\beta,\alpha,a) \quad\quad \wedge \\
Attack(\alpha,\beta,a) \wedge C_{at4} \Rightarrow Stop(\beta,\alpha)
\end{array}
$$

where:

$$
\begin{array}{ll}
C_{at1} & = Op^{at_1}_{at_1} \vee (\neg Op^{at_1}_{at_1} \wedge Op^{at_2}_{at_1}) \\
Op^{at_1}_{at_1} & = Op^{at_1}_{as_1} \\
Op^{at_2}_{at_1} & = Op^{at_2}_{as_1}
\end{array}
$$

$$
\begin{array}{ll}
C_{at2} & = \neg C_{at1} \wedge (Op^{qu_1}_{at_2} \vee (\neg Op^{qu_1}_{at_2} \wedge Op^{qu_2}_{at_2})) \\
Op^{qu_1}_{at_2} & = Op^{qu_1}_{as_2} \\
Op^{qu_2}_{at_2} & = Op^{qu_2}_{as_2}
\end{array}
$$

$$
C_{at3} \quad = \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a \wedge \neg Op^{qu_1}_{at_2} \wedge \neg Op^{qu_2}_{at_2}
$$

$$
\begin{array}{ll}
C_{at4} & = \neg Op^{at_1}_{at_1} \wedge \neg Op^{qu_1}_{at_2} \wedge \neg Op^{qu_2}_{at_2} \wedge \neg C_{at3} \\
& \quad \wedge \forall\, b \in Ar,\ \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \trianglerighteq b \Rightarrow \\
& \quad\quad\quad\quad \Delta(p, \cup CS) = \Delta(p, \cup CS \cup \{b\})
\end{array}
$$

The conditions associated with the Attack rule are similar to the ones defining the Assert rule. The $Attack$ move also includes the case where the agent that initiates the persuasion puts forward a new argument, which is not attacking any existing argument but changing the status of the persuasion topic. This is useful when the advanced arguments cannot be attacked/defended, so that the agent tries another way to convince the addressee.

4- Question Rule

$$\begin{aligned}
&Question(\alpha, \beta, Y) \wedge C_{qu1} \Rightarrow Assert(\beta, \alpha, a) \quad \wedge \\
&Question(\alpha, \beta, Y) \wedge C_{qu2} \Rightarrow Assert(\beta, \alpha, d_{\partial}^{Y'}) \wedge \\
&Question(\alpha, \beta, Y) \wedge C_{qu3} \Rightarrow Assert(\beta, \alpha, ?)
\end{aligned}$$

where:

$$\begin{aligned}
C_{qu1} &= \exists\, a \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a \\
&\wedge (a \uparrow Y \vee a \uparrow \bar{Y})
\end{aligned}$$

$$\begin{aligned}
C_{qu2} &= \exists\, d_{\partial}^{Y'}, d_{\partial}^{Y'}.Y' \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd d_{\partial}^{Y'} \\
&\wedge (d_{\partial}^{Y'} \uparrow Y \vee d_{\partial}^{Y'} \uparrow \bar{Y})
\end{aligned}$$

$$C_{qu23} = \neg C_{qu1} \wedge \neg C_{qu2}$$

Agent β can answer α's question about the content Y by asserting an argument for or against Y. If not, it answers by a partial argument if it can generate it. Otherwise, it answers by "?" which means that it does not know if Y holds or not. We recall that this rule is played when an agent has a partial argument and asks the addressee about the missing assumptions, so that the answer could be the complete missing assumptions, a part of it, or nothing.

5- Stop Rule

$$\begin{aligned}
&Stop(\alpha, \beta) \wedge C_{st1} \Rightarrow Question(\beta, \alpha, Y) \wedge \\
&Stop(\alpha, \beta) \wedge C_{st2} \Rightarrow Stop(\beta, \alpha)
\end{aligned}$$

where:

$$\begin{aligned}
C_{st1} &= Op_{st_1}^{qu_1} \vee (\neg Op_{st_1}^{qu_1} \wedge Op_{st_1}^{qu_2})) \\
Op_{st_1}^{qu_1} &= Op_{as_2}^{qu_1} \\
Op_{st_1}^{qu_2} &= Op_{as_2}^{qu_2}
\end{aligned}$$

$$C_{st2} = \neg C_{st1}$$

Before answering the α's $Stop$ move by another $Stop$ to terminate the protocol, β checks if no other partial arguments changing the status of p could be generated. The

$Stop$ move is played only if no such argument could be generated, which means that the conflict cannot be resolved.

Theorem 1 (Termination). *If $\langle \mathcal{C}, \mathcal{D} \rangle$ is a well-formed persuasion protocol about a wff p, then $\langle \mathcal{C}, \mathcal{D} \rangle$ always terminates either successfully by $Accept$ or unsuccessfully by $Stop$.*

Proof. See Appendix

Definition 19 (Soundness - Completeness). *A persuasion protocol about a wff p is sound and complete iff for some arguments a for or against p we have at the end of the protocol: $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta) \rhd a \Leftrightarrow \mathcal{AR}(\cup CS) \rhd a$.*

Theorem 2 (Soundness and Completeness). *If $\langle \mathcal{C}, \mathcal{D} \rangle$ is a well-formed persuasion protocol about a wff p, then $\langle \mathcal{C}, \mathcal{D} \rangle$ is sound and complete.*

Proof. See Appendix

4 Illustrative Example

This example illustrates a B2B purchase-order scenario involving two businesses (B_1 and B_2). First, a customer places an order for products via `Customer-WS` (WS for Web service). Based on this order, `Customer-WS` obtains details on the customer's purchase history from `CRM-WS` (Customer Relationship Management) of B_1. Afterward, `Customer-WS` forwards these details to B_1's `Billing-WS`, which calculates the customer's bill and sends the bill to `CRM-WS`. This latter prepares the detailed purchase order based on the bill and sends `Inv-Mgmt-WS` (Inventory Management) of B_1 this order for fulfillment. Then, `Inv-Mgmt-WS` sends `Shipper-WS` of B_2 a shipment request. `Shipper-WS` is now in charge of delivering the products to the customer.

The above scenario could be affected by the following conflict: B_2's `Shipper-WS` may not deliver the products as agreed with B_1's `Inv-Mgmt-WS`, perhaps due to lack of trucks. This is a conflict that could be resolved using our protocol by which, `Shipper-WS` tries to persuade `Inv-Mgmt-WS` about the new shipment time and then inform `Customer-WS` of the new delivery time.

Let α_{B_1} be the agent representing `Inv-Mgmt-WS` of B_1 and β_{B_2} be the agent representing `Shipper-WS` of B_2. The resolution of the conflict along with the use of dialogue games are hereafter provided. For the lack of space reason, we will not give the agents' knowledge bases and we will also omit the arguments representation.

1- β_{B_2} identifies the conflict (condition C_{in1} is satisfied) and plays the **Initial game** by asserting an acceptable argument a about lack of trucks from its knowledge base $\mathcal{A}_{\beta_{B_2}}$ supporting its position: $Assert(\beta_{B_2}, \alpha_{B_1}, a)$.

2- α_{B_1} has an argument b attacking β_{B_2}'s argument which is about available trucks committed to others that could be used to ship the products (condition C_{as1} is satisfied). α_{B_1} plays then the **Assertion game** by advancing the $Attack$ move: $Attack(\alpha_{B_1}, \beta_{B_2}, b)$.

3- β_{B_2} replies by playing the **Attack game**. Because it does not have an argument to change the status of the persuasion topic (condition C_{at1} is not satisfied), but has a partial argument for that, which is about the high price of these particular trucks that could be not accepted by α_{B_1} (condition C_{at2} is satisfied), it advances the move: $Question(\beta_{B_2}, \alpha_{B_1}, x)$ where x represents accepting or not the new prices. The idea here is that β_{B_2} can attack α_{B_1}, if it refuses the new prices that others have accepted.

4- α_{B_1} plays the **Question game** and answers the question by asserting an argument c in favor of the increased shipment charges (condition C_{qu1} is satisfied): $Assert(\alpha_{B_1}, \beta_{B_2}, c)$.

5- β_{B_2} plays the **Assertion game**, and from $\mathcal{A}_{\beta_{B_2}} \cup CS_{\alpha_{B_1}}$, it accepts the argument and agrees to deliver the products as per the agreed schedule with the new price, which is represented by d (condition C_{as3} is satisfied): $Accept(\beta_{B_2}, \alpha_{B_1}, d)$. Consequently, the persuasion terminates successfully by resolving the conflict.

5 Related Work

The closest work to the protocol proposed in this paper is the one proposed by [4] for inquiry dialogues. However, there are many fundamental differences between the two protocols. Inquiry and persuasion settings are completely different since the objectives and dynamics of the two dialogues are different. In Black & Hunter's protocol, argumentation is captured only by the notion of argument with no attack relation between arguments. This is because agents collaborate to establish joint proofs. However, in our system, agents can reason about conflicting assumptions, and they should compute different acceptability semantics, not only to win the dispute, but also to reason internally in order to remove inconsistencies from their assumptions. From the specification perspective, there are no similarities between the two protocols. Our protocol is specified as a set of rules about which agents can reason using argumentation, which captures the agents' choices and strategies. However, Black & Hunter's protocol is specified in a declarative manner and the strategy is only defined as a function without specifying how the agents can use it. The adopted moves in the two proposals are also different. Although there is an equivalent notion of partial arguments in Black & Hunter's proposal, the statuses and dynamics we define for these arguments and the fact of considering these arguments as an attempt to change the statuses of uttered arguments are original in our work. Another technical, but fundamental difference in the two protocols is the possibility in our protocol of considering not only the last uttered argument, but any previous argument which allows agents to consider and try different ways of attacking each other.

References

1. Amgoud, L., and Cayrol, C. 1998. On the acceptability of arguments in preferenced-based argumentation framework. In *Proc. of the 14th Conf. on Uncertainty in Art. Int.*, 1–7.
2. Atkinson, K., Bench-Capon, T., and McBurney, P. 2005. A Dialogue Game Protocol for Multi-Agent Argument over Proposals for Action. *J. of AAMAS*, 11(2):153–171.
3. Besnard, P., and Hunter, A. 2001. A logic-based theory of deductive arguments. *J. of Artificial Intelligence*, 128(1-2):203–235.
4. Black, E., and Hunter, A. 2007. A Generative Inquiry Dialogue System. In *The Int. Conf. on Autonomous Agents and Multiagent Systems*, pp. 1010–1017.

5. Bondarenko, A., Dung, P., Kowalski, R.; and Toni, F. 1997. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. *J. of Artificial Intelligence*, 93(1–2):63–101.
6. Brewka, G. 2001. Dynamic Argument Systems: A Formal Model of Argumentation Processes Based on Situation Calculus. *J. of Logic and Computation*, 11(2):257–282.
7. Chesnevar, C., Maguitman, A., and Loui, R. 2000. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383.
8. Dimopoulos, Y., Nebel, B., and Toni, F. 2002. On the computational complexity of assumption-based argumentation for default reasoning. *J. of Artificial Intelligence*, 141(1/2):57-78.
9. Dung, P., Kowalski, R., and Toni, F. 2006. Dialectic proof procedures for assumptionbased, admissible argumentation. *J. of Artificial Intelligence*, 170(2):114-159.
10. Dung, P. 1995. The Acceptability of Arguments and its Fundamental Role in Non-Monotonic Reasoning and Logic Programming and n-Person Game. *J. of Art. Int.*, 77:321–357.
11. Gaertner, D., and Toni, F. 2007. Computing Arguments and Attacks in Assumption-Based Argumentation. *IEEE Intelligent Systems*, 22(6):24–33.
12. Garcia, A., and Simari, G. 2004. Defeasible Logic Programming: an Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138.
13. Kaci, S., and van der Torre, L. 2007. Preference-based Argumentation: Arguments Supporting Multiple Values. *International Journal of Approximate Reasoning*, To appear.
14. Kakas, A., and Toni, F. 1999. Computing argumentation in logic programming. *Journal of Logic Programming*, 9(4):515–562.
15. McBurney, P., and Parsons, S. 2002. Games that Agents Play: A Formal Framework for Dialogues between Autonomous Agents. *Journal of Logic, Language, and Information*, 11(3):315–334.
16. Parsons, S., Wooldridge, M., and Amgoud, L. 2003. On the Outcomes of Formal Inter-Agent Dialogues. In *Proc. of The Int. Conf. on Autonomous Agents and Multiagent Systems*, 616–623.
17. Pasquier, P., Rahwan, I., Dignum, F., and Sonenberg, L. 2006. Argumentation and Persuasion in the Cognitive Coherence Theory. In *The 1st Int. Conf. COMMA*, 223–234. IOS Press.
18. Prakken, H. 2005. Coherence and Flexibility in Dialogue Games for Argumentation. *J. of Logic and Computation*, 15:1009–1040.
19. Rahwan, I., and McBurney, P. 2007. Argumentation Technology. *IEEE Intelligent Systems*, 22(6):21–23.

Appendix

Proof of Proposition 4. *To prove this we should prove that $C_{as3} \Rightarrow \neg C_{as1} \wedge \neg C_{as2}$. Using the logical calculation, we can easily prove that $\neg C_{as1} \wedge \neg C_{as2} = \neg C_{as1} \wedge \neg Op^{qu_1}_{as_2} \wedge \neg Op^{qu_2}_{as_2}$. Also, if an agent β can build an acceptable argument a from $\mathcal{A}_\beta \cup CS_\alpha$, then it cannot build an acceptable or (preferred) semi-acceptable argument attacking a from the same set. Therefore, $\mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a \Rightarrow \neg C_{as1}$. Thus the result follows.* $\Box$

Proof of Theorem 1. *Agents' knowledge bases are finite and repeating moves with the same content is prohibited. Consequently, the number of $Attack$ and $Question$ moves that agents can play is finite. At a given moment, agents will have two possibilities only: $Accept$ if an acceptable argument can be built from $CS_\alpha \cup CS_\beta$, or $Stop$, otherwise. Therefore, the protocol terminates successfully by $Accept$, or unsuccessfully by $Stop$*

when $Accept$ move cannot be played, which means that only semi-acceptable arguments are included in $CS_\alpha \cup CS_\beta$. $\Box$

Proof of Theorem 2. *For simplicity and without loss of generality, we suppose that agent α starts the persuasion.*
Let us first prove the $\Rightarrow$ direction: $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta) \rhd a \Rightarrow \mathcal{AR}(\cup CS) \rhd a$. In the protocol, the persuasion starts when a conflict over p occurs. Consequently, the case where $\mathcal{A}_\alpha \rhd a$ and $\mathcal{A}_\beta \rhd a$ does not hold. Indeed, the possible cases are limited to three:

1. *$\mathcal{A}_\alpha \rhd a$ and $\mathcal{A}_\beta \not\rhd a$. In this case, agent α starts the persuasion over p by asserting a. Agent β can either play the $Attack$ move or the $Question$ move. Because $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta \rhd a)$ all the β's arguments will be counter-attacked. For the same reason, β cannot play the $Stop$ move. Consequently, at the end, β will play an $Accept$ move. It follows that $\mathcal{AR}(\cup CS \rhd a)$.*
2. *$\mathcal{A}_\alpha \not\rhd a$ and $\mathcal{A}_\beta \rhd a$. In this case, agent α starts the persuasion by asserting an acceptable argument b in its knowledge base against p ($\mathcal{A}_\alpha \rhd b$). This argument will be attacked by agent β, and the rest is identical to case 1 by substituting agent roles.*
3. *$\mathcal{A}_\alpha \not\rhd a$ and $\mathcal{A}_\beta \not\rhd a$. To construct argument a out of $\mathcal{A}_\alpha \cup \mathcal{A}_\beta$, two cases are possible. Either, (1) agent α has an acceptable partial argument a_∂^Y for p and agent β has the missing assumptions (or some parts of the missing assumptions, and agent α has the other parts), or (2) the opposite (i.e., agent β has an acceptable partial argument a_∂^Y for p and agent α has the missing assumptions (or some parts of the missing assumptions, and agent β has the other parts)). Only the second case is possible since the first one is excluded by hypothesis. For simplicity, we suppose that agent α has all the missing assumptions, otherwise the missing assumptions will be built by exchanging the different partial arguments. Agent α starts the persuasion by asserting an acceptable argument b in its knowledge base against p. Agent β can either play an $Attack$ or a $Question$ move. If attack is possible, then agent α can either counter-attack or play the $Stop$ move. The same scenario continues until agent α plays $Stop$, and then agent β plays a $Question$ Move. Agent α answers now the question by providing the missing assumptions, after which agent β attacks and agent α can only accept since $\mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta \rhd a)$. It follows that $\mathcal{AR}(\cup CS \rhd a)$.*

Let us now prove the $\Leftarrow$ direction: $\mathcal{AR}(\cup CS) \rhd a \Rightarrow \mathcal{AR}(\mathcal{A}_\alpha \cup \mathcal{A}_\beta) \rhd a$.

In the protocol, to have $\mathcal{AR}(\cup CS) \rhd a$ one of the two agents, say agent α, puts forward the argument a and the other, agent β, accepts it. On the one hand, to advance an argument, agent α plays the $Assert$ move (in the initial or question rules) or $Attack$ move (in the assertion or attack rules). In all these cases, we have: $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a$ and there is no partial acceptable argument attacking a from $\mathcal{A}_\alpha \cup SC_\beta$. On the other hand, to accept an argument (in the assertion or attack rules), agent β should check that $\mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a$, there is no other arguments changing the status of the persuasion topic, and there is no partial acceptable argument attacking a from $\mathcal{A}_\beta \cup SC_\alpha$. Therefore we obtain: $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta \cup \mathcal{A}_\beta \cup CS_\alpha) \rhd a$. Because $CS_\alpha \subseteq \mathcal{A}_\alpha$ and $CS_\beta \subseteq \mathcal{A}_\beta$ we are done. $\Box$