Using Collaborations for Distributed Argumentation with Defeasible Logic Programming

Matthias Thimm
Faculty of Computer Science
Technische Universität Dortmund
Germany

Alejandro J. García
Department of Computer Science and Engineering
Universidad Nacional del Sur, Bahía Blanca
Argentina

Gabriele Kern-Isberner
Faculty of Computer Science
Technische Universität Dortmund
Germany

Guillermo R. Simari
Department of Computer Science and Engineering
Universidad Nacional del Sur, Bahía Blanca
Argentina

Abstract

In this paper, we extend previous work on distributed argumentation using Defeasible Logic Programming. There, several agents form a multi agent setting, in which they are able to generate arguments for a given query and counterarguments to the arguments of other agents. The framework is monitored by a moderator, which coordinates the argumentation process and can be seen as a judge overlooking the defender and accuser in a legal case. We extend this framework by allowing the agents to form alliances. We introduce a notion of cooperation for agents called *collaborations*, which allow the agents not only to argue with one another, but to share their beliefs in order to jointly generate new arguments. We give a declarative definition as well as an algorithmic characterization of the argument generation process and relate our framework with general Defeasible Logic Programming.

Introduction

Defeasible argumentation (Prakken & Vreeswijk 2002) deals with argumentative reasoning using uncertain knowledge. An instantiation of defeasible argumentation is Defeasible Logic Programming (DeLP) by García and Simari (García & Simari 2004) and is an approach for logical argumentative reasoning (Rahwan & Amgoud 2006; Besnard & Hunter 2000) based on defeasible logic. In DeLP the belief in literals is supported by arguments and in order to handle conflicting information a warrant procedure decides which information has the strongest grounds to believe in.

There are many approaches to realize multi agent argumentation and especially negotiation (Kraus 1997; Booth 2002) in multi agent systems. Whereas in (Amgoud, Dimopolous, & Moraitis 2007; Parsons, Sierra, & Jennings 1998) and especially in (Bench-Capon 2003), the focus lies on using argumentation for persuasion, here we use argumentation to reach a common conclusion of a group of agents. Considering a jury court it is reasonable to assume that there are jurors who are less competent in jurisdiction than others. However it is the main goal to reach an agreement regarding the given case rather than unifying the jurors beliefs.

In this paper, a distributed argumentation framework for cooperative agents is introduced in which agents may have independent or overlapping belief bases. Here, following (Thimm 2008; Thimm & Kern-Isberner 2008a), Defeasible Logic Programming is used for knowledge representation. Hence, agents belief bases will be sets of defeasible rules (García & Simari 2004) and agents may build argument using their local rules. Similar to (Móra, Alferes, & Schroeder 1998; de Almeida & Alferes 2006), we will define a notion of *collaboration* and a mechanisms that allow agents to cooperate for building arguments will be introduced.

In many different scenarios the cooperation of agents in a multi agent setting is desirable. Suppose that in a legal dispute a team of lawyers have to work together, acting as one accuser or defender. Or imagine a dispute between political parties, where each member tries to defend their party's interests. The simplest solution to these kinds of scenarios is to represent each whole team or party as one single agent, thus merging the beliefs of the members into one knowledge base. But from a knowledge representational point of view it is more realistic to represent each member of such a team as an individual agent and let these agents collaborate with each other. Another drawback of the first approach is a computational one. If the knowledge of many members of a team is joined, the computation of arguments can be expensive, as the whole knowledge base has to be searched. If a team is made up of many agents, each an expert in his field, the construction and evaluation of arguments can be divided upon them and only the agents, that can contribute, do so.

The framework proposed in (Thimm & Kern-Isberner 2008a) consists of several agents and a central moderator, which coordinates the argumentation process undertaken by the agents. The moderator accepts a query, consisting of a single literal, and asks the agents to argue about the warrant status of it. That framework was motivated for modeling situations where participating agents have opposite views of the given query (e. g. a legal dispute, where agents take the roles of accuser and defender). Therefore, each agent build its own arguments using its local belief and it may attack or defend arguments of other agents. In this paper, we extend that framework by considering groups of agents who may collaborate in order to build better arguments using beliefs of other agents. A collaboration is basically a set of agents that form an alliance for argument construction. With the use of collaborations, we are able to derive more arguments than in the case with no collaborations. The key idea of computing collaborated arguments is similar to (Móra, Alferes, & Schroeder 1998) but uses another concept of a *partial argument*. As will be described below, a partial argument is some kind of an intermediate result when constructing an argument in a distributed manner. Partial arguments help collecting the rules that are necessary to derive a conclusion from the local belief bases.

The paper is organized as follows. In the next section, a brief introduction to Defeasible Logic Programming and the distributed framework of (Thimm & Kern-Isberner 2008a) is presented. We continue by introducing collaborations into the multi agent setting, that allow the agents to jointly build arguments. We proof soundness and completeness of the algorithmic representation of collaborated argument generation, followed by a comparison of our approach with Defeasible Logic Programming and other related work. Finally, we conclude we a summary and an outlook to further work.

Distributed Argumentation using DeLP

We give a brief introduction in the distributed argumentation framework ArgMAS (*Argumentation-based multi agent system*) proposed in (Thimm 2008; Thimm & Kern-Isberner 2008a; 2008b) adapted to our needs in this paper. The framework is based upon DeLP (Defeasible Logic Programming) (García & Simari 2004) and consists of several agents and a central moderator, which coordinates the argumentation process undertaken by the agents. An overview of such a system is depicted in figure 1. The moderator accepts a query, consisting of a single literal, and asks the agents to argue about the warrant status of it, i. e., whether the literal or its negation can be supported by an ultimately undefeated argument. Agents use the global belief base of the system, which contains strict knowledge, and their own local belief bases consisting of defeasible knowledge to generate arguments. Eventually the system returns an answer to the questioner that describes the final status of the literal based on the agents' individual beliefs.

We start our description of this framework by presenting the basic argumentative formalisms of DeLP(García & Simari 2004).

Defeasible Logic Programming

The basic elements of DeLP are facts and rules. Let $\mathcal{L}$ denote a set of ground literals, where a literal h is a ground atom A or a negated ground atom $\sim A$, where the symbol

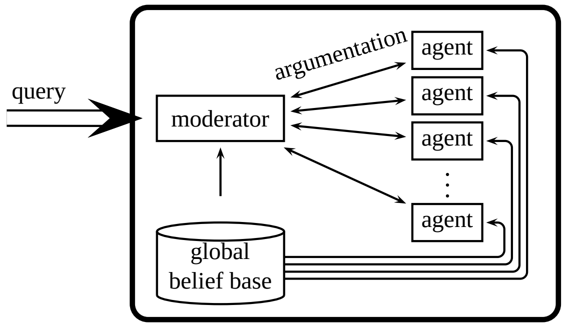

Figure 1: An argumentation-based multi agent system (ArgMAS)

$\sim$ represents the strong negation. Overlining will be used to denote the complement of a literal with respect to strong negation, i. e., it is $\overline{p} = \sim p$ and $\overline{\sim p} = p$ for a ground atom p. A literal $h \in \mathcal{L}$ is also called a *fact*.

The set of rules is divided into strict rules, i. e., rules encoding strict consequences, and defeasible rules which derive uncertain or defeasible conclusions. A *strict rule* is an ordered pair $h \leftarrow B$, where $h \in \mathcal{L}$ and $B \subseteq \mathcal{L}$. A *defeasible rule* is an ordered pair $h \prec B$, where $h \in \mathcal{L}$ and $B \subseteq \mathcal{L}$. A defeasible rule is used to describe tentative knowledge as in "birds fly". We use the functions $body/1$ and $head/1$ to refer to the head resp. body of a defeasible or strict rule. Strict and defeasible rules are ground. However, following the usual convention (Lifschitz 1996), some examples will use "schematic rules" with variables (denoted with an initial uppercase letter). Let DEF_X resp. STR_X be the set of all defeasible resp. strict rules, that can be constructed with literals from $X \subseteq \mathcal{L}$. We will omit the subscripts when referring to the whole set of literals $\mathcal{L}$, e. g. we write DEF for $\mathsf{DEF}_{\mathcal{L}}$.

Using facts, strict and defeasible rules, one is able to derive additional beliefs as in other rule-based systems. Let $X \subseteq \mathcal{L} \cup \mathsf{STR} \cup \mathsf{DEF}$ be a set of facts, strict rules, defeasible rules, and let furthermore $h \in \mathcal{L}$. A *(defeasible) derivation* of h from X, denoted $X \mid\!\sim h$, consists of a finite sequence $h_1, \ldots, h_n = h$ of literals ($h_i \in \mathcal{L}$) such that h_i is a fact ($h_i \in X$) or there is a strict or defeasible rule in X with head h_i and body $b_1, \ldots, b_k$, where every b_l ($1 \leq l \leq k$) is an element h_j with $j < i$. If the derivation of a literal h only uses strict rules, the derivation is called a *strict* derivation. A set X is *contradictory*, denoted $X \mid\!\sim \bot$, iff there exist defeasible derivations of two complementary literals from X. In difference to DeLP, the framework of ArgMAS divides the strict and defeasible knowledge into a global belief base and several local belief bases which constitute the individual beliefs of each agent.

Definition 1 (Belief bases)**.** A *global belief base* $\Pi \subseteq \mathcal{L} \cup \mathsf{STR}$ is a non-contradictory set of strict rules and facts. A set of defeasible rules $\Delta \subseteq \mathsf{DEF}$ is called a *local belief base*.

Given a set of agents $\mathfrak{A} = \{A_1, \ldots, A_n\}$ every agent A_i maintains a local belief base Δ_i ($1 \leq i \leq n$) which represents his own belief.

Example 1 ((García & Simari 2004), example 2.1)**.** Let a global belief base Π and a local belief base Δ be given by

$$\Pi = \left\{ \begin{array}{l} chicken(tina) \\ scared(tina) \\ penguin(tweety) \\ bird(X) \leftarrow chicken(X) \\ bird(X) \leftarrow penguin(X) \\ \sim flies(X) \leftarrow penguin(X) \end{array} \right\},$$

$$\Delta = \left\{ \begin{array}{l} flies(X) \prec bird(X) \\ \sim flies(X) \prec chicken(X) \\ flies(X) \prec chicken(X), scared(X) \\ nests\_in\_trees(X) \prec flies(X) \end{array} \right\}.$$

The global belief base Π contains the facts, that Tina is a scared chicken and that Tweety is penguin. The strict rules state that all chickens and all penguins are birds, and penguins cannot fly. The defeasible rules of the local belief base Δ express that birds normally fly, chickens normally do not fly (except when they are scared) and something that flies normally nests in trees.

As facts and strict rules describe strict knowledge, it is reasonable to assume Π to be non-contradictory, i. e., there are no derivations of complementary literals from Π only. But when considering several (or just one) local belief bases $\Delta_1, \ldots, \Delta_n$ of other agents, which may have different beliefs, then $\Pi \cup \Delta_1 \cup \ldots \cup \Delta_n$ can be contradictory.

Definition 2 (Argument, Subargument)**.** Let $h \in \mathcal{L}$ be a literal and let Π resp. Δ be a global resp. local belief base. $\langle \mathcal{A}, h\rangle$ is an *argument* for h, iff

- $\mathcal{A} \subseteq \Delta$,
- there exists a defeasible derivation of h from $\Pi \cup \mathcal{A}$,
- the set $\Pi \cup \mathcal{A}$ is non-contradictory, and
- $\mathcal{A}$ is minimal with respect to set inclusion.

The literal h will be called *conclusion* and the set $\mathcal{A}$ will be called *support* of the argument $\langle \mathcal{A}, h\rangle$. An argument $\langle \mathcal{B}, q\rangle$ is a *subargument* of an argument $\langle \mathcal{A}, h\rangle$, iff $\mathcal{B} \subseteq \mathcal{A}$. Let $\mathsf{ARG}_{\Pi,\Delta}$ be the set of all arguments that can be built from Π and Δ.

Two literals h and h_1 *disagree* regarding a global belief base Π, iff the set $\Pi \cup \{h, h_1\}$ is contradictory. Two complementary literals p und $\sim p$ disagree trivially, because for every Π the set $\Pi \cup \{p, \sim p\}$ is contradictory. But two literals which are not contradictory, can disagree as well. For $\Pi = \{(\sim h \leftarrow b), (h \leftarrow a)\}$ the literals a and b disagree, because $\Pi \cup \{a, b\}$ is contradictory.

We call an argument $\langle \mathcal{A}_1, h_1\rangle$ a *counterargument* to an argument $\langle \mathcal{A}_2, h_2\rangle$ at a literal h, iff there is a subargument $\langle \mathcal{A}, h\rangle$ of $\langle \mathcal{A}_2, h_2\rangle$ such that h and h_1 disagree.

In order to deal with counterarguments to other arguments, a central aspect of defeasible argumentation becomes a formal comparison criterion among arguments. A possible preference relation among arguments is *Generalized Specificity* (Stolzenburg *et al.* 2003). According to this criterion an argument is preferred to another argument, iff the former one is more *specific* than the latter, i. e., (informally) iff the former one uses more facts or less rules. For example, $\langle\{c \prec a, b\}, c\rangle$ is more specific than $\langle\{\sim c \prec a\}, \sim c\rangle$.

For a formal definition and and desirable properties of preference criterions in general see (Stolzenburg *et al.* 2003; García & Simari 2004). For the rest of this paper we use $\succ$ to denote an arbitrary but fixed preference criterion among arguments. The preference criterion is needed to decide whether an argument defeats another or not, as disagreement does not imply preference.

Definition 3 (Defeater)**.** An argument $\langle \mathcal{A}_1, h_1\rangle$ is a *defeater* of an argument $\langle \mathcal{A}_2, h_2\rangle$, iff there is a subargument $\langle \mathcal{A}, h\rangle$ of $\langle \mathcal{A}_2, h_2\rangle$ such that $\langle \mathcal{A}_1, h_1\rangle$ is a counterargument of $\langle \mathcal{A}_2, h_2\rangle$ at literal h and either $\langle \mathcal{A}_1, h_1\rangle \succ \langle \mathcal{A}, h\rangle$ (*proper defeat*) or $\langle \mathcal{A}_1, h_1\rangle \not\succ \langle \mathcal{A}, h\rangle$ and $\langle \mathcal{A}, h\rangle \not\succ \langle \mathcal{A}_1, h_1\rangle$ (*blocking defeat*).

When considering sequences of arguments, the definition of defeat is not sufficient to describe a conclusive argumentation line. Defeat only takes an argument and its counterargument into consideration, but disregards preceeding arguments. But we expect also properties like *non-circularity* or *concordance* from an argumentation sequence. See (García & Simari 2004) for a more detailed description of acceptable argumentation lines.

Definition 4 (Acceptable Argumentation Line)**.** Let Π be a global belief base. Let $\Lambda = [\langle \mathcal{A}_1, h_1\rangle, \ldots, \langle \mathcal{A}_m, h_m\rangle]$ be a sequence of some arguments. Λ is called *acceptable argumentation line*, iff

1. Λ is a finite sequence,
2. every argument $\langle \mathcal{A}_i, h_i\rangle$ with $i > 1$ is a defeater of its predecessor $\langle \mathcal{A}_{i-1}, h_{i-1}\rangle$ and if $\langle \mathcal{A}_i, h_i\rangle$ is a blocking defeater of $\langle \mathcal{A}_{i-1}, h_{i-1}\rangle$ and $\langle \mathcal{A}_{i+1}, h_{i+1}\rangle$ exists, then $\langle \mathcal{A}_{i+1}, h_{i+1}\rangle$ is a proper defeater of $\langle \mathcal{A}_i, h_i\rangle$,
3. $\Pi \cup \mathcal{A}_1 \cup \mathcal{A}_3 \cup \ldots$ is non-contradictory (*concordance of supporting arguments*),
4. $\Pi \cup \mathcal{A}_2 \cup \mathcal{A}_4 \cup \ldots$ is non-contradictory (*concordance of interfering arguments*), and
5. no argument $\langle \mathcal{A}_k, h_k\rangle$ is a subargument of an argument $\langle \mathcal{A}_i, h_i\rangle$ with $i < k$.

Let SEQ denote the set of all sequences of arguments that can be built using rules from DEF, STR and facts from $\mathcal{L}$.

Let $+$ denote the concatenation of argumentation lines and arguments, e. g. $[\langle \mathcal{A}_1, h_1\rangle, \ldots, \langle \mathcal{A}_n, h_n\rangle] + \langle \mathcal{B}, h\rangle$ stands for $[\langle \mathcal{A}_1, h_1\rangle, \ldots, \langle \mathcal{A}_n, h_n\rangle, \langle \mathcal{B}, h\rangle]$.

In DeLP a literal h is *warranted*, if there is an argument $\langle \mathcal{A}, h\rangle$ which is non-defeated in the end. To decide whether $\langle \mathcal{A}, h\rangle$ is defeated or not, every acceptable argumentation line starting with $\langle \mathcal{A}, h\rangle$ has to be considered.

Definition 5 (Dialectical Tree)**.** Let Π be a global belief base and $\Delta_1, \ldots, \Delta_n$ be local belief bases. Let $\langle \mathcal{A}_0, h_0\rangle$ be an argument. A *dialectical tree* for $\langle \mathcal{A}_0, h_0\rangle$, denoted $\mathcal{T}_{\langle \mathcal{A}_0, h_0\rangle}$, is defined as follows.

1. The root of $\mathcal{T}$ is $\langle \mathcal{A}_0, h_0\rangle$.
2. Let $\langle \mathcal{A}_n, h_n\rangle$ be a node in $\mathcal{T}$ and let $\Lambda = [\langle \mathcal{A}_0, h_0\rangle, \ldots, \langle \mathcal{A}_n, h_n\rangle]$ be the sequence of nodes from the root to $\langle \mathcal{A}_n, h_n\rangle$. Let $\langle \mathcal{B}_1, q_1\rangle, \ldots, \langle \mathcal{B}_k, q_k\rangle$ be the defeaters of $\langle \mathcal{A}_n, h_n\rangle$. For every defeater $\langle \mathcal{B}_i, q_i\rangle$ with $1 \leq i \leq k$ such that the argumentation line $\Lambda' =$

$[\langle\mathcal{A}_0, h_0\rangle, \ldots, \langle\mathcal{A}_n, h_n\rangle, \langle\mathcal{B}_i, q_i\rangle]$ is acceptable, the node $\langle\mathcal{A}_n, h_n\rangle$ has a child $\langle\mathcal{B}_i, q_i\rangle$. If there is no such $\langle\mathcal{B}_i, q_i\rangle$, the node $\langle\mathcal{A}_n, h_n\rangle$ is a leaf.

Let DIA denote the set of all dialectical trees with arguments that can be built using rules from DEF, STR and facts from $\mathcal{L}$.

In order to decide whether the argument at the root of a given dialectical tree is defeated or not, it is necessary to perform a *bottom-up*-analysis of the tree. Every leaf of the tree is marked "undefeated" and every inner node is marked "defeated", if it has at least one child node marked "undefeated". Otherwise it is marked "undefeated". Let $\mathcal{T}^*_{\langle\mathcal{A},h\rangle}$ denote the marked dialectical tree of $\mathcal{T}_{\langle\mathcal{A},h\rangle}$.

We call a literal h *warranted*, iff there is an argument $\langle\mathcal{A}, h\rangle$ for h such that the root of the marked dialectical tree $\mathcal{T}^*_{\langle\mathcal{A},h\rangle}$ is marked "undefeated". Then $\langle\mathcal{A}, h\rangle$ is a *warrant* for h. Observe that, if a literal h is a fact or has a strict derivation from a the global belief base Π alone, then h is also warranted as there are no counterarguments for $\langle\emptyset, h\rangle$.

Formal Description of the Distributed Framework

We now describe the components of the distributed framework, namely the moderator and the agents, using a functional description of their intended behaviour. As the framework of ArgMAS is flexible, many different definitions of the functions to be presented can be thought of. But we restrain them on the notions of DeLP as described above, so we use the subscript "D" to denote the DeLP specific implementation.

When the moderator receives arguments from the agents, he builds up several dialectical trees and finally he has to evaluate them using the bottom-up evaluation method described above.

Definition 6 (Analysis function χ_{D})**.** The *analysis function* χ_{D} is a function $\chi_{\mathsf{D}} : \mathsf{DIA} \rightarrow \{0, 1\}$ such that for every dialectical tree $\upsilon \in \mathsf{DIA}$ it holds $\chi_{\mathsf{D}}(\upsilon) = 1$ iff the root argument of υ is undefeated.

Furthermore the evaluation of dialectical trees makes only sense, if the tree was built up according to the definition of an acceptable argumentation line. Hence, the moderator and the agents as well, have to check whether new arguments are valid in the current argumentation line.

Definition 7 (Acceptance function $\eta_{\mathsf{D},\succ}$)**.** For a given preference relation $\succ$ among arguments, the *acceptance function* $\eta_{\mathsf{D},\succ}$ is a function $\eta_{\mathsf{D},\succ} : \mathsf{SEQ} \rightarrow \{0, 1\}$ such that for every argument sequence $\Lambda \in \mathsf{SEQ}$ it holds $\eta_{\mathsf{D},\succ}(\Lambda) = 1$ iff Λ is acceptable according to Definition 4.

It is possible to assume different acceptance functions for different agents according to different definitions of an acceptable argumentation line (Thimm & Kern-Isberner 2008b). But in our multi agent system, we assume $\eta_{\mathsf{D},\succ}$ to be fixed and the same for the moderator and all agents by convention.

At the end of the argumentation process for a query h, the agents have produced a set of dialectical trees with root arguments for h or $\overline{h}$, respectively. As we have to distinguish several different cases, the moderator has to decide, whether the query h is warranted, the negation of h is warranted, or none of them are warranted in the framework. Let $\mathfrak{P}(S)$ denote the power set of a set S.

Definition 8 (Decision function μ_{D})**.** The *decision function* μ_{D} is a function $\mu_{\mathsf{D}} : \mathfrak{P}(\mathsf{DIA}) \rightarrow \{\text{YES,NO,UNDECIDED,UNKNOWN}\}$. Let $Q_{\dot{p}} \subseteq \mathsf{DIA}$ such that all root arguments of dialectical trees in $Q_{\dot{p}}$ are arguments for p or for $\overline{p}$, then μ_{D} is defined as

1. $\mu_{\mathsf{D}}(Q_{\dot{p}}) = \text{YES}$, if there is a dialectical tree $\upsilon \in Q_{\dot{p}}$ s. t. the root of υ is an argument for p and $\chi_{\mathsf{D}}(\upsilon) = 1$.
2. $\mu_{\mathsf{D}}(Q_{\dot{p}}) = \text{NO}$, if there is a dialectical tree $\upsilon \in Q_{\dot{p}}$ s. t. the root of υ is an argument for $\overline{p}$ and $\chi_{\mathsf{D}}(\upsilon) = 1$.
3. $\mu_{\mathsf{D}}(Q_{\dot{p}}) = \text{UNDECIDED}$, if $\chi_{\mathsf{D}}(\upsilon) = 0$ for all $\upsilon \in Q_{\dot{p}}$.
4. $\mu_{\mathsf{D}}(Q_{\dot{p}}) = \text{UNKNOWN}$, if p is not in the language ($p \notin \mathcal{L}$).

The function μ_{D} is well-defined, as it cannot be the case that both conditions 1. and 2. are simultaneously fulfilled, see for example (Thimm & Kern-Isberner 2008c).

The above functions are sufficient to define the moderator of the framework.

Definition 9 (Moderator)**.** For a given preference relation $\succ$ among arguments, the *moderator* is a tuple $(\mu_{\mathsf{D}}, \chi_{\mathsf{D}}, \eta_{\mathsf{D},\succ})$.

The agents of the framework provide two functionalities. First, they propose initial arguments for a given literal (or its negation) submitted by the moderator of the framework, which will be roots of the dialectical trees to be constructed. For a given query h it may be necessary to examine both, all dialectical trees with a root argument for h and all dialectical trees with a root argument for $\overline{h}$, as a query for h can only be answered with NO if there is a warrant for $\overline{h}$. Second, the agents propose counterarguments to arguments of other agents[1] that are valid in the given argumentation line. An agent is not obliged to return all his valid arguments for a given query or all his counterarguments for a given argument. Therefore, it is possible to model different kinds of argumentation strategies given different instantiations of the following argument functions.

Definition 10 (Root argument function)**.** Let Π be a global belief base and let Δ be a local belief base. A *root argument function* $\varphi_{\Pi,\Delta}$ relative to Π and Δ is a function $\varphi_{\Pi,\Delta} : \mathcal{L} \rightarrow \mathfrak{P}(\mathsf{ARG}_{\Pi,\Delta})$ such that for every literal $h \in \mathcal{L}$ the set $\varphi_{\Pi,\Delta}(h)$ is a set of arguments for h or for $\overline{h}$ from Π and Δ.

Definition 11 (Counterargument function)**.** Let Π be a global belief base and let Δ be a local belief base. A *counterargument function* $\psi_{\Pi,\Delta}$ relative to Π and Δ is a function $\psi_{\Pi,\Delta} : \mathsf{SEQ} \rightarrow \mathfrak{P}(\mathsf{ARG}_{\Pi,\Delta})$ such that for every argumentation sequence $\Lambda \in \mathsf{SEQ}$ the set $\psi_{\Pi,\Delta}(\Lambda)$ is a set of attacks from Π and Δ on the last argument of Λ and for every $\langle\mathcal{B}, h\rangle \in \psi_{\Pi,\Delta}(\Lambda)$ it holds that $\eta_{\mathsf{D},\succ}(\Lambda + \langle\mathcal{B}, h\rangle) = 1$.

[1]Furthermore the agents can possibly propose counterarguments to their own arguments, but here we will not consider this case explicitly.

Here we assume that the root argument and counterargument functions of all agents are the same and especially *complete*, i. e, they return all possible arguments for the given situation and do not omit one.

Given the above definitions an agent of the framework is defined as follows.

Definition 12 (Agent)**.** Let Π be a global belief base. An *agent* relative to Π is a tuple $(\Delta, \varphi_{\Pi,\Delta}, \psi_{\Pi,\Delta})$ with a local belief base Δ relative to Π, a root argument function $\varphi_{\Pi,\Delta}$ and a counterargument function $\psi_{\Pi,\Delta}$.

Finally, the definition of an argumentation-based multi agent system can be given as follows.

Definition 13 (Argumentation-based multi agent system)**.** An *argumentation-based multi agent system* (ArgMAS) is a tuple $(M, \Pi, \{A_1, \dots, A_n\})$ with a moderator M, a global belief base Π and agents $A_1, \dots, A_n$ relative to Π.

Given an ArgMAS T and a query h, the framework produces an answer to h as follows. First, the moderator of T asks all agents for initial arguments for h and for $\overline{h}$ and starts a dialectical tree with each of them as root arguments. Then for each of these arguments, the moderator asks every agent for counterarguments and incorporates them into the corresponding dialectical trees accordingly. This process is repeated for every new argument until no more arguments can be constructed. Eventually the moderator analyses the resulting dialectical trees and returns the appropriate answer to the questioner. A dialectical tree built via this process is called an *argumentation product*. The answer behaviour of an ArgMAS is determined by the decision function of its moderator. For a query $h \in \mathcal{L}$ and and ArgMAS T the *answer* of T on h is $\mu_{\mathsf{D}}(\{\upsilon_1, \dots, \upsilon_n\})$, where μ_{D} is the decision function of the moderator of T and $\{\upsilon_1, \dots, \upsilon_n\}$ is the set of all argumentation products of T for h.

We conclude this section with an example that illustrates the above definitions.

Example 2. Suppose an ArgMAS $T = (M, \Pi, \{A_1, A_2\})$ with two agents A_1, A_2. The global belief base Π and the local belief bases Δ_1 resp. Δ_2 of the agents A_1 resp. A_2 are given by

$$\begin{aligned} \Pi &= \{a, b\}, \\ \Delta_1 &= \{(d \mathrel{-\!\!\prec} a, c), (c \mathrel{-\!\!\prec} b)\}, \\ \Delta_2 &= \{(\sim\!c \mathrel{-\!\!\prec} a, b)\}. \end{aligned}$$

Assume *Generalized Specificity* as the preference relation among arguments and let d be the query under consideration. When the moderator passes this query to the agents, only the root argument function of A_1 returns a non-empty set of root arguments, namely the set that contains the one argument $X_1 = \langle\{(d \mathrel{-\!\!\prec} a, c), (c \mathrel{-\!\!\prec} b)\}, d\rangle$. The moderator starts the construction of one dialectical tree with X_1 as its root. Then he asks every agent for counterarguments on X_1 that are acceptable after the argumentation line $[X_1]$. There only the counterargument function of A_2 returns the only possible counterargument $\langle\{(\sim\!c \mathrel{-\!\!\prec} a, b)\}, \sim\!c\rangle$. After that, no more arguments can be constructed and the final dialectical tree, i. e., the one final argumentation product υ can be seen in Figure 2. After applying his analysis function, the moderator determines that the root argument of υ is marked "defeated" and as υ is the only argumentation product of T on d, the answer of the decision function of the moderator and thus the answer of the system on d is UNDECIDED.

$$\langle\{(d \mathrel{-\!\!\prec} a, c), (c \mathrel{-\!\!\prec} b)\}, d\rangle$$

$$\langle\{(\sim\!c \mathrel{-\!\!\prec} a, b)\}, \sim\!c\rangle$$

Figure 2: The one argumentation product υ in Example 2.

Collaborations

The distributed argumentation framework described above serves well when modeling scenarios, where the agents are involved in some kind of a dispute and have opposite views of the given query, such as a legal dispute, where agents take the roles of accuser and defender (Thimm 2008). But the framework fails to model situations, in which the agents should cooperate in order to reach a common solution, because they cannot share their beliefs in order to construct arguments that cannot be constructed by one agent alone.

Example 3. Let $T = (M, \Pi, \{A_1, A_2\})$ be an ArgMAS with $\Pi = \{a, d\}$ and let Δ_1 resp. Δ_2 be the local belief bases of A_1 resp. A_2 with

$$\begin{aligned} \Delta_1 &= \{(b \mathrel{-\!\!\prec} a), (b \mathrel{-\!\!\prec} a, c)\} \quad \text{and} \\ \Delta_2 &= \{(\sim\!b \mathrel{-\!\!\prec} a), (c \mathrel{-\!\!\prec} d)\} \quad . \end{aligned}$$

Given the query b, T yields two argumentation products

$$\begin{aligned} &[\langle\{(b \mathrel{-\!\!\prec} a)\}, b\rangle, \langle\{(\sim\!b \mathrel{-\!\!\prec} a)\}, \sim\!b\rangle] \quad \text{and} \\ &[\langle\{(\sim\!b \mathrel{-\!\!\prec} a)\}, \sim\!b\rangle, \langle\{(b \mathrel{-\!\!\prec} a)\}, b\rangle] \quad . \end{aligned}$$

As the roots of both argumentation products will be marked "defeated", the answer of T on b is UNDECIDED.

Observe that there is the additional argument $\langle\{(b \mathrel{-\!\!\prec} a, c), (c \mathrel{-\!\!\prec} d)\}, b\rangle$, that could be constructed, if both agents share their beliefs. This argument cannot be defeated by $\langle\{(\sim\!b \mathrel{-\!\!\prec} a)\}, \sim\!b\rangle$, as the first is more specific than the second, and thus would be a warrant for b. So in this case, the answer of the system for query b should be YES instead of UNDECIDED.

In (Móra, Alferes, & Schroeder 1998) a framework for cooperating agents in a context very similar to that of an ArgMAS was introduced. There – in contrast to here – extended logic programs (Gelfond & Lifschitz 1991) were used to model an agent's belief. We follow the ideas of (Móra, Alferes, & Schroeder 1998) to define the notion of *collaboration* and the mechanisms that allow our agents to cooperate in an ArgMAS, but extend their framework according to our needs.

We begin by defining a *collaboration* which describes a coalition of several agents, like a team of lawyers or a political party. A collaboration describes a set of agents each obliged to one another to support them with necessary information.

Definition 14 (Collaboration)**.** Let $T = (M, \Pi, \{A_1, \ldots, A_n\})$ be an ArgMAS. A *collaboration* C of T is a set of agents with $C \subseteq \{A_1, \ldots, A_n\}$.

A collaboration is basically a set of agents that form an alliance for argument construction. With the use of collaborations, we are able to derive more arguments than in the case with no collaborations. Observe, that we do not impose any conditions on collaborations. Although it might be appropriate to enforce the agents in a collaboration (for example) to have non-conflicting beliefs, we do not restrain the above definition to stay simple in our presentation. Furthermore, if an agent has conflicting beliefs with its partners in a collaboration, this does not affect the conjoint construction of arguments, since not all rules of each agents have to be in conflict.

Definition 15 (Collaborated argument)**.** Let $T = (M, \Pi, \mathfrak{A})$ be an ArgMAS and $C = \{A_1, \ldots, A_l\} \subseteq \mathfrak{A}$ a collaboration of T. If $\Delta_1, \ldots, \Delta_l$ are the local belief bases of $A_1, \ldots, A_l$, then an argument $\langle \mathcal{A}, h\rangle$ is a *collaborated argument* of the collaboration C iff $\langle \mathcal{A}, h\rangle \in \mathsf{ARG}_{\Pi, \Delta_1 \cup \ldots \cup \Delta_l}$, i. e., $\langle \mathcal{A}, h\rangle$ is an argument regarding the global belief base Π with $\mathcal{A} \subseteq \Delta_1 \cup \ldots \cup \Delta_l$.

Example 4. Consider again the ArgMAS of example 3. Suppose agents A_1 and A_2 are members of a collaboration C, i. e., $C = \{A_1, A_2\}$. Then

$$\langle \{(b \prec a, c), (c \prec d)\}, b\rangle$$

is a collaborated argument of C.

We call $\langle \mathcal{A}, h\rangle$ a *strict* collaborated argument of the collaboration C iff it is a collaborated argument of C and it can not be constructed by any agent alone, i. e., it is $\mathcal{A} \nsubseteq \Delta$ for every local belief base Δ of an agent in C. For instance, the argument in example 4 is a strict collaborated argument. In the upcoming algorithm, we do not intend to generate only strict collaborated arguments. This means, that the algorithm will also generate arguments, that could have been generated by an agent alone. A modification of the algorithm to suppress the generation of non-strict collaborated arguments is straightforward, but loses simplicity and clarity.

We can describe the intended behaviour of the distributed framework including collaborations by introducing meta agents, each representing a collaboration, and then subsuming the extended case with collaborations by the simple framework described in the last section. Without considering these meta agents, the generation of collaborated arguments must be done completely autonomously by the agents of a collaboration alone. We do not address this issue in the present work, but leave it open for future research. For now, assume that φ_C^{coll} resp. ψ_C^{coll} is a root argument resp. counterargument function that generates collaborated arguments of the collaboration C. We will give a formal definition of these functions and an operational description of their computation in the next subsection.

Definition 16 (Associated meta agent)**.** Let $T = (M, \Pi, \mathfrak{A})$ be an ArgMAS, $C = \{A_1, \ldots, A_l\} \subseteq \mathfrak{A}$ be a collaboration of T with $\Delta_1, \ldots, \Delta_l$ being the local belief bases of agents $A_1, \ldots, A_l \in \mathfrak{A}$ and η be an acceptance function. The agent $(\emptyset, \varphi_C^{coll}, \psi_C^{coll}, \eta)$ is called the *meta agent associated to the collaboration* C with functions $\varphi_C^{coll} : \mathcal{L} \rightarrow \mathfrak{P}(\mathsf{ARG}_{\Pi, \Delta_1 \cup \ldots \cup \Delta_l})$ and $\psi_C^{coll} : \mathsf{SEQ} \rightarrow \mathfrak{P}(\mathsf{ARG}_{\Pi, \Delta_1 \cup \ldots \cup \Delta_l})$.

Definition 17 (Collaborative ArgMAS)**.** A tuple $T = (M, \Pi, \{A_1, \ldots, A_n\}, \{C_1, \ldots, C_m\})$ is a *collaborative ArgMAS* if $T' = (M, \Pi, \{A_1, \ldots, A_n, A_{C_1}, \ldots, A_{C_m}\})$ is an ArgMAS with A_{C_i} being the meta agent associated to the collaboration C_i (for $1 \leq i \leq m$).

The above definition does not impose, that an agent cannot belong to more than one collaboration, but in the following we only consider the case, where $C_1, \ldots, C_m$ are disjoint.

Before turning to the operational aspects of computing collaborated arguments, we give a small example to illustrate collaborations.

Example 5. Let $\Pi = \{(h \leftarrow a, b), c, d\}$ and two local belief bases Δ_1, Δ_2 of two agents A_1, A_2 given by

$$\begin{aligned} \Delta_1 &= \{(a \prec c), (g \prec d)\}, \\ \Delta_2 &= \{(b \prec f), (f \prec g)\}. \end{aligned}$$

Let $C = \{A_1, A_2\}$ be a collaboration and A_C the corresponding meta agent. When asked for an argument for h the two agents alone A_1 and A_2 can obviously not return any. But when combing their beliefs, the meta agent A_C is able to generate the argument

$$\langle \{(a \prec c), (b \prec f), (f \prec g), (g \prec d)\}, h\rangle \quad ,$$

which makes also use of the strict rule $h \leftarrow a, b$.

Generating collaborated arguments

The key idea of computing collaborated arguments is similar to (Móra, Alferes, & Schroeder 1998) but uses another characterization of a *partial argument*. While Móra et al. impose a partial argument to be a partial derivation with no intermediate rules missing, we define a partial argument declaratively as an argument with some additional facts missing. For both, a partial argument is some kind of an intermediate result when constructing an argument in a distributed manner.

Definition 18 (Partial argument)**.** Let Π be a global belief base and $R \subseteq \mathsf{DEF}$ a set of defeasible rules. A tuple $\langle \mathcal{A}, h\rangle$ is a *partial argument* for a literal h regarding Π and R, iff $\mathcal{A} \subseteq R$ and there is a set of literals $F \subseteq \mathcal{L}$ such that $\langle \mathcal{A}, h\rangle \in \mathsf{ARG}_{\Pi \cup F, R}$, i. e., $\langle \mathcal{A}, h\rangle$ is an argument in $(\Pi \cup F, R)$. The smallest sets F (regarding set inclusion) satisfying this condition are called *free sets*. The set of all free sets is denoted $free(\langle \mathcal{A}, h\rangle)$ for a partial argument $\langle \mathcal{A}, h\rangle$. Let $\mathsf{PAR}_{\Pi, R}$ be the set of all partial arguments for the global belief base Π and a set of defeasible rules $R \subseteq \mathsf{DEF}$.

Example 6. Let $\Pi = \{(h \leftarrow a), (h \leftarrow b)\}$. Then $\langle \emptyset, h\rangle$ is a partial argument (regarding DEF), since there is a set of literals, namely $\{a\}$, such that $\langle \emptyset, h\rangle$ is an argument regarding $\Pi' = \{(h \leftarrow a), (h \leftarrow b), a\}$. The same is true for the set $\{b\}$, so the free sets of $\langle \emptyset, h\rangle$ regarding Π are given by

$$free(\langle \emptyset, h\rangle) = \{\{a\}, \{b\}\} \quad .$$

Example 7. Let $\Pi = \{(h \leftarrow a, b), b\}$ and $\mathcal{A} = \{(a \prec c)\}$. Then $\langle \mathcal{A}, h\rangle$ is a partial argument (regarding DEF), since $\{c\}$ is a free set of $\langle \mathcal{A}, h\rangle$ regarding Π:

$$free(\langle \mathcal{A}, h\rangle) = \{\{c\}\}$$

Observe, that every argument $\langle \mathcal{A}, h\rangle$ is also a partial argument (with $free(\langle \mathcal{A}, h\rangle) = \emptyset$), as well as $\langle \emptyset, h\rangle$ for any h.

Our approach to compute collaborated arguments is a top-down approach that starts with the empty set and iteratively adds defeasible rules until the given conclusion can be derived. For this purpose we equip every agent with a function that extends a given partial argument as much as possible.

Definition 19 (Partial argument function)**.** Let Π be a global belief base, A be an agent and Δ_A its local belief base. A *partial argument function* κ_A for agent A is a function $\kappa_A : \mathsf{PAR}_{\Pi,\mathsf{DEF}} \rightarrow \mathfrak{P}(\mathsf{PAR}_{\Pi,\mathsf{DEF}})$ and is defined as

$$\kappa_A(\langle \mathcal{A}, h\rangle) = \{\langle \mathcal{A}', h\rangle \in \mathsf{PAR}_{\Pi,\mathcal{A}\cup\Delta_A} \mid \mathcal{A}' \supset \mathcal{A}\}$$

Example 8. Let $\Pi = \{(d \leftarrow e)\}$ be a global belief base and A be an agent with a local belief base

$$\Delta = \{(g \prec c, d), (e \prec f)\} \quad .$$

Then it is

$$\kappa_A(\langle \emptyset, g\rangle) = \{\langle \mathcal{A}_1, g\rangle, \langle \mathcal{A}_2, g\rangle\}$$

with

$$\begin{aligned} \mathcal{A}_1 &= \{(g \prec c, d)\}, \\ \mathcal{A}_2 &= \{(g \prec c, d), (e \prec f)\}. \end{aligned}$$

Furthermore it is $free(\langle \mathcal{A}_1, g\rangle) = \{\{c, d\}, \{c, e\}\}$ and $free(\langle \mathcal{A}_2, g\rangle) = \{\{c, f\}\}$

Using the partial argument functions of the agents in a collaboration, the associated meta agent is able to compute the collaborated arguments for a given literal h with Algorithm 1. The algorithm `CollaboratedArguments` takes as input a global belief base Π, a collaboration of agents $\{A_1, \ldots, A_l\}$ and a literal h, and it returns the set of all collaborated arguments of $\{A_1, \ldots, A_l\}$ in a backward chaining manner.

```
1   CollaboratedArguments(Π,{A_1,...,A_l},h)
2     iArgs := {⟨∅,h⟩}
3     cArgs := ∅
4     while iArgs ≠ ∅
5       remove a tuple ⟨A,h⟩ from iArgs
6       if free(⟨A,h⟩) = ∅ then
7         cArgs := cArgs ∪ ⟨A,h⟩
8       else
9         for i from 1 to l
10          iArgs := iArgs ∪ κ_{A_i}(⟨A,h⟩)
11    return cArgs
```

Algorithm 1: Construction of collaborated arguments

First, the algorithm initializes the set of partial arguments `iArgs` with the trivial partial argument $\langle \emptyset, h\rangle$ (line 2). As long as there are partial arguments available, the algorithm removes one of them from `iArgs` and extends it in every possible way, i. e., by eliminating free literals from any free set by every agents' partial argument function. When an argument is complete, i. e., $free(\langle \mathcal{A}, h\rangle) = \emptyset$, the argument can be added to the result set (line 6,7).

Example 9. Let Π be a global belief base with

$$\Pi \quad = \quad \{(g \leftarrow c), (d \leftarrow f), a, b\}$$

and let A_1, A_2 be two agents with local belief bases Δ_1, Δ_2, respectively, given by

$$\begin{aligned} \Delta_1 &= \{(c \prec a), (c \prec h), (d \prec e)\} \quad \text{and} \\ \Delta_2 &= \{(g \prec d), (e \prec b)\} \quad . \end{aligned}$$

Let g be a query and consider the following exemplary execution of `CollaboratedArguments` on the call `CollaboratedArguments(`Π`,`$\{A_1, A_2\}$`,`g`)`.

First, the set `iArgs` is initialized with the partial argument $X_1 = \langle \emptyset, g\rangle$. As $free(X_1) = \{\{g\}, \{c\}\} \neq \emptyset$ the algorithm continues at line 10. There, $\kappa_{A_1}(X_1)$ is called yielding $\{X_2, X_3\}$ as the set of possible extensions to X_1 with $X_2 = \langle (c \prec a), g\rangle$ and $X_3 = \langle (c \prec h), g\rangle$. Then $\kappa_{A_2}(X_1)$ is called yielding $X_4 = \langle (g \prec d), g\rangle$ as a possible extension to X_1. So back at line 4 we have `iArgs`$= \{X_2, X_3, X_4\}$.

Let then X_3 be chosen at line 5. As $free(X_3) = \{\{h\}\}$ the algorithm continues at line 10. Neither agent can extend X_3 because neither has a defeasible rule with head h nor is there a strict rule with head h, so it is $\kappa_{A_1}(X_3) = \kappa_{A_2}(X_3) = \emptyset$. Back at line 4 we have `iArgs`$= \{X_2, X_4\}$.

Let then X_2 be chosen at line 5. As $free(X_2) = \emptyset$ the algorithm continues at line 7 and X_2 is added to the result set `cArgs`.

Now it is `iArgs`$= \{X_4\}$ and $free(X_4) = \{\{d\}\}$. Continuing at line 10, $\kappa_{A_1}(X_4)$ is called yielding $X_5 = \langle (g \prec d), (d \prec e), g\rangle$ and $\kappa_{A_2}(X_4)$ is called yielding no further extension to X_4. So back at line 4 we have `iArgs`$= \{X_5\}$ with $free(X_5) = \{\{e\}\}$. Finally, agent A_2 completes X_5 yielding $X_6 = \langle (g \prec d), (d \prec e), (e \prec b), g\rangle$ with $free(X_6) = \emptyset$.

So, `CollaboratedArguments(`Π`,`$\{A_1, A_2\}$`,`g`)` returns the set `cArgs`$= \{X_2, X_6\}$.

Using the `CollaboratedArguments` algorithm we are now able to define the root argument and counterargument functions of the associated meta agents.

Definition 20 (Root argument function φ_C^{coll})**.** Let Π be a global belief base, $C = \{A_1, \ldots, A_l\}$ a collaboration with Δ_i being the local belief base of agent A_i ($1 \leq i \leq l$) and h a literal. The function $\varphi_C^{coll} : \mathcal{L} \rightarrow \mathfrak{P}(\mathsf{ARG}_{\Pi,\Delta_1\cup\ldots\cup\Delta_l})$ is defined as

$$\begin{aligned} \varphi_C^{coll}(h) \quad = \quad & \texttt{CollaboratedArguments}(\Pi, C, h) \cup \\ & \texttt{CollaboratedArguments}(\Pi, C, {\sim}h) \end{aligned}$$

Definition 21 (Counterargument function ψ_C^{coll})**.** Let Π be a global belief base, C $C = \{A_1, \ldots, A_l\}$ a collaboration with Δ_i being the local belief base of agent A_i ($1 \leq i \leq l$) and h a literal. Let λ be an argumentation sequence. The function $\psi_C^{coll} : \mathsf{SEQ} \rightarrow \mathfrak{P}(\mathsf{ARG}_{\Pi,\Delta_1\cup\ldots\cup\Delta_l})$ is defined as

$$\begin{aligned} \psi_C^{coll}(\lambda) = \{ & \langle \mathcal{A}, h\rangle \in \mathsf{ARG}_{\Pi,\Delta_1\cup\ldots\cup\Delta_l} \mid \exists h : \langle \mathcal{A}, h\rangle \in \\ & \texttt{CollaboratedArguments}(\Pi, C, h) \wedge \\ & \lambda + \langle \mathcal{A}, h\rangle \text{ is acceptable } \} \end{aligned}$$

Example 10. We continue Example 5. So let $\Pi = \{(h \leftarrow a, b), c, d\}$ and two local belief bases Δ_1, Δ_2 of two agents A_1, A_2 given by

$$\Delta_1 = \{(a \prec c), (g \prec d)\},$$
$$\Delta_2 = \{(b \prec f), (f \prec g)\}.$$

Let there be a collaboration $C = \{A_1, A_2\}$ and A_C the corresponding meta agent. Given the query h, the agent A_C would use his root argument function φ_C^{coll} and thus the algorithm `CollaboratedArguments` in order to generate a root argument for or against h. In the algorithm `CollaboratedArguments` for the literal h the set `iArgs` is initialized with $\{\langle\emptyset, h\rangle\}$ with $free(\langle\emptyset, h\rangle) = \{\{h\}, \{a, b\}\}$. This means, that the partial argument $\langle\emptyset, h\rangle$ can be completed by either an argument for h or by arguments for both a and b using the strict rule $(h \leftarrow a, b)$. Observe that there is no possibility to complete $\langle\emptyset, h\rangle$ without the use of the strict rule $(h \leftarrow a, b)$, as no agent has a defeasible rule with h as its head.

Next, suppose, that A_C asks agent A_1 to extend the partial argument $\langle\emptyset, h\rangle$. As A_1 has a defeasible rule for a, he can extend $\langle\emptyset, h\rangle$ to $\langle\mathcal{A}, h\rangle$ with $\mathcal{A} = \{(a \prec c)\}$ and $free(\langle\mathcal{A}, h\rangle) = \{\{b\}\}$. Agent A_2 can then extend $\langle\mathcal{A}, h\rangle$ to $\langle\mathcal{A}', h\rangle$ with $\mathcal{A}' = \{(a \prec c), (b \prec f), (f \prec g)\}$ and $free(\langle\mathcal{A}', h\rangle) = \{\{g\}\}$ and finally agent A_1 can extend $\langle\mathcal{A}', h\rangle$ to $\langle\mathcal{A}'', h\rangle$ with $\mathcal{A}'' = \{(a \prec c), (b \prec f), (f \prec g), (g \prec d)\}$ and $free(\langle\mathcal{A}'', h\rangle) = \emptyset$.

Observe, that the algorithm `CollaboratedArguments` generates this argument also on other ways than the described above, e. g. by first using the partial argument $\langle(b \prec f), h\rangle$ provided by A_2.

Soundness and completeness

We will now show, that the algorithm `CollaboratedArguments` is sound and complete. The soundness and completeness of the root argument function φ_C^{coll} and the counterargument function ψ_C^{coll} then follow directly.

We start by showing soundness, i. e., that every argument $\langle\mathcal{A}, h\rangle$ that is returned by `CollaboratedArguments(`Π`,`$\{A_1, \ldots, A_n\}$`,`h`)` is indeed a collaborated argument of C with conclusion h.

Theorem 1 (Soundness)**.** *Let Π be a global belief base, $C = \{A_1, \ldots, A_n\}$ be a collaboration of agents $A_1, \ldots, A_n$ with local belief bases $\Delta_1, \ldots, \Delta_n$ respectively. If h is a literal and*

$$\langle\mathcal{A}, h'\rangle \in \texttt{CollaboratedArguments(}\Pi\texttt{,}C\texttt{,}h\texttt{)},$$

then $h = h'$ and $\langle\mathcal{A}, h'\rangle$ is a collaborated argument of C.

Proof. It is clear due to line 7 of Algorithm 1 that $h = h'$. So it remains to show, that $\langle\mathcal{A}, h\rangle$ is a collaborated argument of C, i. e., that $\langle\mathcal{A}, h\rangle$ is an argument of (Π, Δ') with $\Delta' = \Delta_1 \cup \ldots \cup \Delta_n$.

1. Clearly it is $\mathcal{A} \subseteq \Delta'$ because the partial argument function of an agent A_i ($1 \leq i \leq n$) only adds defeasible rules to the argument that belong to $\Delta_i \subseteq \Delta'$.
2. $\langle\mathcal{A}, h\rangle$ defeasibly derives h, because it is $free(\langle\mathcal{A}, h\rangle) = \emptyset$ (line 6 in Algorithm 1).
3. $\mathcal{A}$ is non-contradictory, because the partial argument functions of the agents only return partial arguments due to definition. A partial argument must be non-contradictory, as there must be an extension (possibly an extension by $\emptyset$ as in the last completion step of a partial argument) that is an argument and hence non-contradictory.
4. $\langle\mathcal{A}, h\rangle$ is minimal using the same argumentation as above.

□

Furthermore our algorithm is complete in the sense, that if $\langle\mathcal{A}, h\rangle$ is a collaborated argument of a collaboration C with respect to a global belief base Π, then $\langle\mathcal{A}, h\rangle$ will be returned by the algorithm `CollaboratedArguments`.

Theorem 2 (Completeness)**.** *Let Π be a global belief base, $C = \{A_1, \ldots, A_n\}$ be a collaboration of agents $A_1, \ldots, A_n$ with local belief bases $\Delta_1, \ldots, \Delta_n$ respectively. If $\langle\mathcal{A}, h\rangle$ is a collaborated argument of C then it is*

$$\langle\mathcal{A}, h\rangle \in \texttt{CollaboratedArguments(}\Pi\texttt{,}C\texttt{,}h\texttt{)} \quad .$$

Proof. We have to show, that $\langle\mathcal{A}, h\rangle$ is added to the set `cArgs` at line 7 of Algorithm 1. If h can be strictly derived from Π, i. e., it is $\mathcal{A} = \emptyset$, then it is $free(\langle\mathcal{A}, h\rangle) = \emptyset$ and $\langle\mathcal{A}, h\rangle$ is added at line 7 of Algorithm 1. Otherwise, as $\langle\mathcal{A}, h\rangle$ is a collaborated argument of C, there is a defeasible rule $r \in \mathcal{A}$ with $head(r) \in K$ for some $K \in free(\langle\emptyset, h\rangle)$. Let $r \in \Delta_k$ for some $k \in \{1, \ldots, n\}$, then agent A_k will extend the partial argument $\langle\emptyset, h\rangle$ with at least rule r in line 10 of Algorithm 1. Inductively it follows that there is always an extension of this argument by rules of $\mathcal{A}$. As $\langle\mathcal{A}, h\rangle$ is an argument, it is $free(\langle\mathcal{A}, h\rangle) = \emptyset$ and so $\langle\mathcal{A}, h\rangle$ is added to `cArgs` in line 7 of Algorithm 1. □

Related work and comparison

In (Thimm & Kern-Isberner 2008b) it has been shown, that the distributed framework without collaborations subsumes ordinary DeLP, as every defeasible logic program can be translated into an equivalent distributed framework with the same answer behaviour. The other way round is not always possible, as there are distributed settings, where there is no equivalent single defeasible logic program that models the same situation. With the use of collaborations we are now able to establish an equivalence between a special case of the distributed framework with collaborations and ordinary DeLP. For the special case of a collaborative ArgMAS with one collaboration involving all agents, the answer behaviour is the same as when considering a defeasible logic program which is built upon the union of all local belief bases.

Many other proposals exist for introducing argumentative capabilities into distributed systems and especially negotiation systems, see for example (Kraus 1997; Amgoud, Dimopolous, & Moraitis 2007; Bench-Capon 2003; Rueda, Garcia, & Simari 2002; Karunatillake *et al.* 2005).

There are especially two other approaches, that have similarities with the approach proposed in this paper. The framework of (Móra, Alferes, & Schroeder 1998; de Almeida & Alferes 2006) uses extended logic programs to model an agent's belief and defines a notion of distributed argumentation using these extended logic programs. The framework uses the argumentation semantics from (Prakken 1997) and defines a notion of cooperation, that allows the agents to share their beliefs in order to construct new arguments. As this framework uses extended logic programs as the underlying representation formalism, it has a declarative semantics in contrast to the dialectical semantics of DeLP used here.

Black et al. (Black 2007; Black & Hunter 2007) also use defeasible logic programming as the underlying representation formalism to model distributed argumentation. Complementary to the proposal in this paper, the focus of (Black 2007) is on modeling communication protocols and strategies for successful argumentation between agents. They introduce two kinds of inquiry dialogues, one to generate combined arguments and one for the actual argumentation.

Conclusion

Usually, argumentation is considered as a dialectical process which involves two parties, a proponent and an opponent who generate arguments in order to evaluate reasons in favor of or against claims. Argumentation might even reflect deliberations taking place within one single agent.

In this paper, we study argumentation in distributed scenarios in which the pro and con parties consist of several collaborating agents, each agent possessing its own subjective beliefs but sharing strict knowledge with all other agents. As a proper framework to realize such distributed argumentation, we choose DeLP (García & Simari 2004) since it allows a distinction between strict, commonly known world knowledge, on one side, and subjective and defeasible beliefs, on the other. Via collaborations, the agents may produce more and better arguments as any of them might bring forward when only using its own belief base. For each collaboration, we introduce a meta agent that organizes the generation of arguments and counterarguments from the rule reservoir of each agent in a dialogue. As a crucial concept for handling fragments of arguments effectively to build complete arguments, we defined *partial arguments* by modifying an idea from (Móra, Alferes, & Schroeder 1998).

For the operational part of our approach, we present an algorithm to generate collaborated arguments, and prove its soundness and completeness. Finally, we show that the results in this paper generalize the approach proposed in (Thimm & Kern-Isberner 2008a), and compare our work to related approaches.

As part of our ongoing work, we explore different applications of our collaborative argumentation framework. One particularly appealing scenario is to realize negotiations in a multi-agent system under confidentiality constraints. In such scenario, each agent tries to hide its subjective beliefs as well as possible, while at the same time being interested in making as much information available as necessary to reach a good negotiation result.

Acknowledgments The authors thank the reviewers for their very helpful comments to improve the original version of this paper.

References

Amgoud, L.; Dimopolous, Y.; and Moraitis, P. 2007. A unified and general framework for argumentation-based negotiation. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agents Systems, AAMAS'2007*.

Bench-Capon, T. 2003. Persuasion in practical argument using value based argumentation frameworks. *Journal of Logic and Computation* 13(3):429–448.

Besnard, P., and Hunter, A. 2000. Towards a logic-based theory of argumentation. In *Proc. of the 17th American Nat. Conf. on Artif. Intelligence (AAAI'2000)*, 411–416.

Black, E., and Hunter, A. 2007. A generative inquiry dialogue system. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07)*. IEEE Press.

Black, E. 2007. *A Generative Framework for Argumentation-Based Inquiry Dialogues*. Ph.D. Dissertation, University College London.

Booth, R. 2002. Social contraction and belief negotiation. In *Proceedings of the Eighth Conference on Principles of Knowledge Representation and Reasoning (KR 2002)*, 375–384.

de Almeida, I. C., and Alferes, J. J. 2006. An argumentation-based negotiation for distributed extended logic programs. In *Proceedings of CLIMA VII*, 191–210.

García, A., and Simari, G. 2004. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming* 4(1-2):95–138.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385.

Karunatillake, N. C.; Jennings, N. R.; Rahwan, I.; and Norman, T. J. 2005. Argument-based negotiation in a social context. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 1331–1332. New York, NY, USA: ACM.

Kraus, S. 1997. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence* 94(1-2):79–97.

Lifschitz, V. 1996. Foundations of logic programming. In *Principles of Knowledge Representation*. CSLI Publications. 69–127.

Móra, I. A.; Alferes, J. J.; and Schroeder, M. 1998. Argumentation and cooperation for distributed extended logic programs. In *Nonmonotonic Reasoning Workshop'98*.

Parsons, S.; Sierra, C.; and Jennings, N. 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3):261–292.

Prakken, H., and Vreeswijk, G. 2002. Logical systems for defeasible argumentation. In Gabbay, D., and Guenthner, F., eds., *Handbook of Philosophical Logic*, volume 4. Dordrecht: Kluwer Academic Publishers, 2 edition. 219–318.

Prakken, H. 1997. Dialectical proof theory for defeasible argumentation with defeasible priorities (preliminary report). In *ModelAge Workshop*, 202–215.

Rahwan, I., and Amgoud, L. 2006. An argumentation-based approach for practical reasoning. In Weiss, G., and Stone, P., eds., *5th International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS'2006*, 347–354.

Rueda, S. V.; Garcia, A.; and Simari, G. R. 2002. Argument-based negotiation among BDI agents. *Journal of Computer Science and Technology* 2(7).

Stolzenburg, F.; García, A.; Chesnevar, C. I.; and Simari, G. 2003. Computing generalized specificity. *Journal of Non-Classical Logics* 13(1):87–113.

Thimm, M., and Kern-Isberner, G. 2008a. A distributed argumentation framework using defeasible logic programming. In Besnard, P.; Doutre, S.; and Hunter, A., eds., *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA'08)*, number 172 in Frontiers in Artificial Intelligence and Applications, 381–392. Toulouse, France: IOS Press.

Thimm, M., and Kern-Isberner, G. 2008b. A distributed argumentation framework using defeasible logic programming (extended version). Technical report, Technische Universität Dortmund.

Thimm, M., and Kern-Isberner, G. 2008c. On the relationship of defeasible argumentation and answer set programming. In Besnard, P.; Doutre, S.; and Hunter, A., eds., *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA'08)*, number 172 in Frontiers in Artificial Intelligence and Applications, 393–404. Toulouse, France: IOS Press.

Thimm, M. 2008. *Verteilte logikbasierte Argumentation: Konzeption, Implementierung und Anwendung im Rechtswesen*. VDM Verlag Dr. Müller.