An Argumentation-Based Negotiation for Distributed Extended Logic Programs*

Iara Carnevale de Almeida[1,2] and José Júlio Alferes[1]

[1] CENTRIA, Universidade Nova de Lisboa
2829-516 Caparica, Portugal
{ica|jja}@di.fct.unl.pt
[2] Department of Computer Science, Universidade de Évora
Colégio Luis Verney; 7000-671 Évora, Portugal
ica@di.uevora.pt

Abstract. The paradigm of argumentation has been used in the literature to assign meaning to knowledge bases in general, and logic programs in particular. With this paradigm, rules of a logic program are viewed as encoding arguments of an agent, and the meaning of the program is determined by those arguments that somehow (depending on the specific semantics) can defend themselves from the attacks of other arguments.

Most of the work on argumentation-based logic programs semantics has focused on assigning meaning to single programs. In this paper we propose an argumentation-based negotiation semantics for distributed knowledge bases represented as extended logic programs that extends the existing ones by considering sets of (distributed) logic programs, rather than single ones. For specifying the ways in which the various logic programs may combine their knowledge we make use of concepts that had been developed in the areas of defeasible reasoning, distributed knowledge bases, and multi-agent setting. In particular, we associate to each program P a cooperation set (the set of programs that can be used to complete the knowledge in P) and the argumentation set (the set of programs with which P has to reach a consensus).

1 Introduction

The ability to view logic programming as a non-monotonic knowledge representation language brought to light the importance of defining clear declarative semantics for logic programs, for which proof procedures (and attending implementations) are then defined. This work is by now well established and consolidated in what concerns the semantics of single logic programs, in particular approaches where the argumentation metaphor is used for providing such clear declarative semantics e.g. [6,11,3,16,15,10]. This metaphor seems adequate too for modelling situations where there are distributed logic programs, each with some 'knowledge about the world', that might negotiate in order to determine the truth value of common conclusions. However, the previous mentioned works do not directly address this issue.

* The work was partially supported by the Brazilian CAPES, and by the European Commission within the 6th Framework Programme project REWERSE, number 506779.

K. Inoue, K. Satoh, and F. Toni (Eds.): CLIMA VII, LNAI 4371, pp. 191–210, 2007.

In this paper we propose an argumentation-based negotiation semantics for sets of knowledge bases distributed through a multi-agent setting (MAS). In it different agents may have independent or overlapping knowledge bases Kb, each Kb being represented by an extended logic program with denials. If all such agents have complete access to the knowledge bases of all other agents, then they should be able to build arguments using rules of others (cooperate) and would have to defend their arguments against arguments build by the others (argue). In this case, the semantics of argumentation-based negotiation framework should coincide with the semantics of the union of the knowledge bases, viewed as a single one. Here we want to deal with cases where the semantics of multi setting does not necessarily coincide with the union. The basis of our proposal is that agents negotiate by exchanging parts of their knowledge to obtain a consensus concerning the inference of an agent's beliefs. Furthermore, our proposal allows modelling of multi setting with different kinds of purposes. For instance each agent may represent "acquired knowledge" in different periods of time, and we want to know the truth value of some agent's belief in a specific period of time. Another example is when the whole set represents a kind of hierarchy of knowledge such as an organisation where each agent has incomplete (or partial) knowledge of the overall process. Yet another, when we want to "organise" the knowledge about taxonomy into different agents and so represent their natural relation of preferences.

Moreover, a multi-agent setting $\mathcal{A}$ might have the agent's knowledge base physically distributed over a computer network. Therefore, an agent Ag of $\mathcal{A}$ does not need to, and sometimes cannot, argue and/or to cooperate with all agents in $\mathcal{A}$. In our proposal, we state that every agent Ag in $\mathcal{A}$ has associated two sets of agents: the set of agents with which it can cooperate in order to build arguments, and the set of agents with which it must defend from attacks (argue) in order to reach some consensus. In general, little is assumed about these sets: we only impose that every agent argues and cooperates with itself because it would make little sense for an agent neither to access its own knowledge nor to obtain a consensus based upon its own knowledge.

The ability of associating these sets to each agents provides a flexible framework which, besides reflecting the possibly existing physical network, may serve for different purposes as the ones above. For example, for modelling knowledge over a hierarchy where each node of the hierarchy is represented by a Kb that cooperates with all its inferiors, and must argue with all its superiors. Another example is modelling knowledge that evolves. Here the "present" can use knowledge from the "past" unless this knowledge from the past is in conflicting with later knowledge. This can be modelled by allowing any present node to cooperate with its past nodes, and forcing any past node to argue with future nodes. In all these cases, it is important that the knowledge is not flattened, as in the union of all knowledge bases, and that the semantics is parametric on the specific Kb. I.e. it might happens that an argument is acceptable in a given (agent) Kb, and not acceptable in another (agent) Kb of the same system.

As with other argumentation based frameworks (e.g. the ones mentioned above) the semantics is defined based on a notion of acceptable arguments, this notion being itself based on an attacking relation among arguments. Moreover, as in [12], based on acceptability, all arguments are assigned a status: roughly, justified argument are those that are always acceptable; overruled arguments are those that are attacked by a justified

argument; other arguments (that may or may not be acceptable but are not attacked by a justified one) are called defensible.

It is also a goal of the proposed framework to be able to deal with mutually inconsistent, and even inconsistent, knowledge bases. Moreover, when in presence of contradiction we want to obtain ways of multi-agent setting reasoning, ranging from consistent (in which inconsistencies lead to no result) to paraconsistent. For achieving this, the agents may exchange strong or weak arguments, as it is made clear below. This also yield a refinement of the possible status of arguments: justified arguments may now be contradictory, based on contradiction or non-contradictory.

In the next section we define the declarative semantics of the proposed framework. The paper continues by showing some properties of the framework, namely properties that relate it to extant work on argumentation and on other semantics for logic programs. We then present a couple of illustrative examples, and end with some conclusions. For lack of space all proofs have been removed from this version of the paper.

2 Declarative Semantics

As motivated in the introduction, in our framework the knowledge bases of agents are modelled by logic programs. More precisely, we use *Extended Logic Program with denials*, itself an extension of Extended Logic Programs [7], for modelling the knowledge bases. Formally:

Definition 1 (Language). *An alphabet $\mathcal{B}$ of a language $\mathcal{L}$ is a finite disjoint set of constants and predicate symbols. Moreover, the symbol $\bot \notin \mathcal{B}$.*

An atom *over $\mathcal{B}$ is an expression of the form $p(t_1, \ldots, t_n)$ where p is a predicate symbol of $\mathcal{B}$ and the t_i's are terms. A* term *over $\mathcal{B}$ is either a variable or a constant. An* objective literal *over $\mathcal{B}$ is either an atom A or its explicit negation $\neg A$. A* default literal *over $\mathcal{B}$ is of the form $not\ A$ where A is an objective literal. A* literal *is either an objective literal or a default literal. By $not\ \{L_1, \ldots, L_n\}$ we mean the set of default literals $\{not\ L_1, \ldots, not\ L_n\}$. By (negative) hypothesis of an objective literal L we mean $not\ L$. By explicit complement of an objective literal L we mean $\neg L$ if L is an atom, or A if $L = \neg A$.*

A term (resp. atom, literal) is called ground *if it does not contain variables. By the Extended Herbrand Base $\mathcal{H}$ of $\mathcal{B}$, $\mathcal{H}(\mathcal{B})$, we mean the set of all ground objective literals of $\mathcal{B}$.*

Definition 2 (Extended Logic Program with Denials). *An* extended logic program with denials (ELPd) over a language $\mathcal{L}$ *is a (finite) set of (ground) rules of the form*

$$L_0 \leftarrow L_1, \ldots, L_l, not\ L_{l+1}, \ldots, not\ L_n\ (0 \leq l \leq n)$$

or of denials of the form

$$\bot \leftarrow L_1, \ldots, L_l, not\ L_{l+1}, \ldots, not\ L_n\ (0 \leq l \leq n)$$

where each L_i $(0 \leq i \leq n)$ is an objective literal of $\mathcal{L}$. A rule is ground if all literals are ground. As usual L_0 is called the head*, and $L_1, \ldots, not\ L_n$ the* body *of the rule. If $n = 0$ the rule is called a* fact *and the arrow symbol is omitted.*

Besides the knowledge base, in our framework each argumentative agent Ag in a multi-agent setting $\mathcal{A}$ must have a unique identity of Ag in $\mathcal{A}$, and two sets of agents' identifiers corresponding to argumentative and cooperative agents with Ag, respectively. Moreover, the identity of Ag is in both sets of argumentative and cooperative agents with Ag:

Definition 3 (Argumentative Agent). *An* argumentative agent *(or* agent*, for short) over a language $\mathcal{L}$ and a set of identifiers Ids is a tuple*

$$Ag =< \alpha,\ Kb_\alpha,\ \mathcal{A}rgue_\alpha,\ \mathcal{C}ooperate_\alpha >$$

where $\alpha \in Ids$, Kb_α is an ELPd over $\mathcal{L}$, $\mathcal{A}rgue_\alpha \subseteq Ids$ and $\mathcal{C}ooperate_\alpha \subseteq Ids$ such that $\alpha \in \mathcal{A}rgue_\alpha$ and $\alpha \in \mathcal{C}ooperate_\alpha$.

We denote by $Id(Ag)$ (resp. $Kb_{Id(Ag)}$, $\mathcal{A}rgue_{Id(Ag)}$ and $\mathcal{C}ooperate_{Id(Ag)}$), the 1st (resp. 2nd, 3rd and 4th) position of the tuple Ag, and by $\mathcal{H}(Id(Ag))$ the set of all atoms and explicitly negated atoms of $Kb_{Id(Ag)}$.

Hereafter, we say 'arguments from $\mathcal{C}ooperate_{Id(Ag)}$ (or $\mathcal{A}rgue_{Id(Ag)}$)' instead of 'arguments from agents whose identities are in $\mathcal{C}ooperate_{Id(Ag)}$ (or $\mathcal{A}rgue_{Id(Ag)}$)'.

Definition 4 (Multi-agent argumentative setting). *Let $\mathcal{L}$ be a language, and Ids be a set of identifiers. A* Multi-Agent argumentative setting *(or* Multi-Agent setting*, for short) $\mathcal{A}$ is a set of agents*

$$\mathcal{A} = \{Ag_1, \ldots, Ag_n\}$$

such that all of the Ag_is are agents over $\mathcal{L}$ and Ids, and no two Ag_is have the same identifier. The Extended Herbrand Base of $\mathcal{A}$, $\mathcal{H}(\mathcal{A})$, is the union of all $\mathcal{H}(\alpha_i)$ such that $\alpha_i \in Ids$.

An *argument*, of an agent Ag, for some objective literal L is a *complete well-defined sequence* concluding L over the *set of rules* of Ag's knowledge base. A *complete* sequence of rules means that all required rules are in the sequence. A *well-defined sequence* means a (minimal) sequence of rules concluding some L. For dealing with consistent and paraconsistent reasoning, we define strong and weak arguments, based on strong and weak sets of rules, the former being simply the rules in the Kbs of agents. A *weak set of rules* results from adding to rule bodies the default negation of the head's complement, and of $\bot$, thus making the rules weaker (more susceptible to being contradicted/attacked).

Definition 5 (Strong and Weak Sets of rules). *Let $\mathcal{L}$ be a language, and P be an ELPd over $\mathcal{L}$. The* strong set of rules of P *is $R^s_P = P$ and the* weak set of rules of P *is*

$$R^w_P = \{\ L \leftarrow Body, not\ \neg L, not\ \bot \mid L \leftarrow Body \in P\ \}$$

We say R_P is a set of rules*, if it is either a strong or a weak set of rules of P.*

A well-defined sequence for an objective literal L is then built by chaining rules as follows: the head of the last rule in the chain is L; furthermore, if some atom L' (ignoring default literals) appears in the body of a rule then there must be a rule before this one with L' in the head; moreover, the sequence must not be circular and only use rules that are strictly necessary.

Definition 6 (Well-defined Sequence). *Let P be an ELPd, and $L \in \mathcal{H}(P)$. A* well-defined sequence *for L over a set of (ground) rules S is a finite sequence $[r_1; \ldots; r_m]$ of rules r_i from S of the form $L_i \leftarrow Body_i$ such that:*

- *L is the head of the rule r_m, and*
- *an objective literal L' is the head of a rule r_i ($1 \leq i < m$) only if L' is not in the body of any r_k ($1 \leq k \leq i$) and L' is in the body of some r_j ($i < j \leq m$)*

We say that a well-defined sequence for L is complete *if for each objective literal L' in the body of the rules r_i ($1 \leq i \leq m$) there is a rule r_k ($k < i$) such that L' is the head of r_k.*

Since we are concerned with modelling knowledge bases distributed over a multi-agent setting, partial arguments of Ag for L must be considered. In fact, an agent alone might not have in its knowledge base enough rules to form a complete argument, but may have part of an argument (a partial argument) that can be complete with knowledge from other agents with which it is able to cooperate. By a partial argument of Ag for L we mean a non-complete well-defined sequence for L, called Seq_L, over the set of rules of Ag's knowledge base. The (both complete and partial) arguments of Ag built only with its own rules are called local arguments of Ag. Since we want to deal with local partial arguments, an argumentation-based semantics with cooperation is proposed. By *argumentation*, we mean the evaluation of arguments to obtain a consensus about a common knowledge; by *cooperation*, we mean the granting of arguments to achieve knowledge completeness.

Definition 7 (Local (Partial or Complete) Argument). *Let $\mathcal{A}$ be a MAS, Ag be an agent in $\mathcal{A}$, $\alpha = Id(Ag)$, Kb_α be the ELPd of Ag, R^s_α (resp. R^w_α) be the strong (resp. weak) set of rules of Kb_α, and $L \in \mathcal{H}(\mathcal{A})$.*

A strong *(resp.* weak*)* local partial argument of α *for L is a pair (α, Seq_L) such that Seq_L is a well-defined sequence for L over R^s_α (resp. R^w_α). A* strong *(resp.* weak*)* local complete argument of α *for L is any strong (resp. weak) partial local argument (α, Seq_L) such that Seq_L is complete and non-empty. We say that (α, Seq_L) is a k-*local argument of α *for L, $LA^k_\alpha(L)$, if it is either a local partial argument or a local complete argument over R^k_L of α for L (where k is either s, for strong arguments, or w, for weak ones).*

The *set of local arguments* of Ag contains all possible local (complete and partial) arguments over Ag's knowledge base.

Definition 8 (Set of Local Arguments). *Let $\mathcal{A}$ be a MAS, and α be an agent's identity in $\mathcal{A}$. The set of k-local arguments of α is:*

$$LA^k_{\mathcal{A}}(\alpha) = \bigcup_{L \in \mathcal{H}(\mathcal{A})} LA^k_\alpha(L)$$

where $LA^s_\alpha(L_i)$ (resp. $LA^w_\alpha(L_i)$) is the set of all strong (resp. weak) local arguments of α for L_i. Local arguments *of α are*

$$LA_{\mathcal{A}}(\alpha) = LA^s_{\mathcal{A}}(\alpha) \cup LA^w_{\mathcal{A}}(\alpha)$$

and we denote by $LA(\mathcal{A})$ the union of all local arguments of agents in $\mathcal{A}$.

Note that an agent Ag in a multi-agent setting $\mathcal{A}$ is able to build arguments for an objective literal L in $\mathcal{H}(\mathcal{A})$ even when Ag has no knowledge about such L (i.e. there is no rule $L \leftarrow Body$ in $Kb_{Id(Ag)}$). This is so because empty sequences are not ruled out by Definition 7. Now, partial arguments of an agent may be completed with the "help" of a sets of (complete and partial) arguments from $\mathcal{C}ooperate_{Id(Ag)}$.

To complete a local partial argument of an agent Ag with (partial or complete) arguments from $\mathcal{C}ooperate_{Id(Ag)}$, we need first to define an *operator* to concatenate these arguments in terms of well-defined sequences[1].

Definition 9 (Operator $+$). *Let $1 \leq i \leq n$, Seq_i is a well-defined sequence for an objective literal L_i, and R_i be the set of all rules in Seq_i.*

The concatenation $Seq_1 + \ldots + Seq_n$ is the set of all well-defined sequences for L_n over $\bigcup_{i=1}^{n} R_i$.

We introduce cooperation by defining a *set of available arguments* of an agent Ag given a set S of (complete or partial) arguments. Every (complete or partial) argument of Ag in S is considered an available argument of Ag. Moreover, if a partial argument for an objective literal L of Ag may be further completed with arguments in S belonging to $\mathcal{C}ooperate_{Id(Ag)}$, this further completed argument is also available.

Definition 10 (Set of Available Arguments). *Let $\mathcal{A}$ be a MAS and α be an agent's identity in $\mathcal{A}$. The set of available arguments given a set S of arguments, $Av(S)$, is the least set such that:*

- *if $(\alpha, Seq_L) \in S$ then $(\alpha, Seq_L) \in Av(S)$, and*
- *if $\exists\{(\beta_1, Seq_{L'}), \ldots, (\beta_i, Seq_L)\} \subseteq Av(S)$ and $\{\beta_1, \ldots, \beta_i\} \subseteq \mathcal{C}ooperate_\alpha$ then for any $NSeq_L \in Seq_{L'} + \ldots + Seq_L$, $(\alpha, NSeq_L) \in Av(S)$*

where $\alpha, \beta_1, \ldots, \beta_i$ are agent's identifiers in $\mathcal{A}$. Let $LA(\mathcal{A})$ be the set of local arguments of $\mathcal{A}$. We denote by $\mathcal{A}rgs(\mathcal{A})$ the set of available arguments of $\mathcal{A}$ *given $LA(\mathcal{A})$, and dub it the* set of all available arguments in *$\mathcal{A}$. Members of $\mathcal{A}rgs(\mathcal{A})$ will be called* arguments.

As mentioned in the Introduction, we are concerned with obtaining ways of reasoning in a multi-agent setting, ranging from consistent to paraconsistent. For this, the agents cooperate and argue by exchanging strong and weak arguments. We assume that every proponent (resp. opponent) agent in a given multi-agent setting exchanges arguments in the same way, i.e. every proposed (resp. opposing) argument is a strong or weak argument. The following two properties reinforce such an assumption. According to the first property, every available argument is of the same kind, strong or weak, as the given set of arguments. From the second property, we see that an agent might have all arguments from its cooperative agents.

Proposition 1. *If S is a set of strong (resp. weak) arguments, then $Av(S)$ is also a set of strong (resp. weak) arguments.*

[1] In short, several distinct well-defined sequences are obtained when concatenating two or more well-defined sequences. Furthermore, we can obtain well-defined sequences that are not in fact complete.

Proposition 2. *Any available argument* (β, Seq_L) *of* β *for* L *is an available argument* (α, Seq_L) *of* α *for* L *iff* $\beta \in \mathcal{C}ooperate_\alpha$.

The following example illustrates how available arguments are built via operator $+$. This example also depicts available arguments built by agents that are not directly interrelated, i.e. there is an "indirect cooperation" between such agents.

Example 1. Let be $\mathcal{A} = \{Ag_1, Ag_2, Ag_3\}$ such that each agent is

$$\begin{array}{rcl} Ag_1 & = & < 1, \{a \leftarrow b\},\ \{1\},\ \{1\}\ > \\ Ag_2 & = & < 2, \{c \leftarrow not\ b\},\ \{2\}, \{1,2\}\ > \\ Ag_3 & = & < 3, \{b; d \leftarrow not\ a\},\ \{3\}, \{2,3\}\ > \end{array}$$

The set of strong local arguments of $\mathcal{A}$ is

$$LA^s(\mathcal{A}) = \left\{ \begin{array}{c} (1,[]), (1,[a \leftarrow b]), \\ (2,[]), (2,[c \leftarrow not\ b]), \\ (3,[]), (3,[b]), (3,[d \leftarrow not\ a]) \end{array} \right\}$$

For simplicity, we call $LA^s(\mathcal{A})$ as S. Based on the first condition of Definition 10, every argument in S is an available argument, i.e. $Av(S) = S$. Based on the second condition of Definition 10, we further obtain the following available arguments:

- $(1, [c \leftarrow not\ b])$ because $\{(2, [c \leftarrow not\ b]), (1, [])\} \subset Av(S)$;
- since $\{(3, [b]), (2, [])\} \subset Av(S)$, $(2, [b]) \in Av(S)$.
  Similarly $(2, [d \leftarrow not\ a]) \in Av(S)$, because $\{(3, [d \leftarrow not\ a]), (2, [])\} \subset Av(S)$;
- as a consequence, $(1, [b])$ and $(1, [d \leftarrow not\ a])$ are available arguments because $\{(2, [b]), (1, [])\} \subset Av(S)$ and $\{(2, [d \leftarrow not\ a]), (1, [])\} \subset Av(S)$, respectively;
- because $\{(1, [a \leftarrow b]), (1, [b])\} \in Av(S)$, $(1, [b;\ a \leftarrow b]) \in Av(S)$ [2].

The least set of available arguments of $\mathcal{A}$ given $LA^s(\mathcal{A})$ is

$$\begin{array}{rl} Av(LA^s(\mathcal{A})) = LA^s(\mathcal{A}) \cup \{ & (1, [c \leftarrow not\ b]),\ (2, [b]),\ (2, [d \leftarrow not\ a]), \\ & (1, [b]),\ (1, [d \leftarrow not\ a]),\ (1, [b; a \leftarrow b])\ \} \end{array}$$

From Definition 10, $Args(\mathcal{A})$ contains all available arguments of an agent Ag built via cooperation with agents in $\mathcal{C}ooperate_{Id(Ag)}$. However, some of these arguments might not be acceptable with respect to arguments in $Args(\mathcal{A})$ from $\mathcal{A}rgue_{Id(Ag)}$. We now describe how a negotiation process should be performed, where cooperation and argumentation are interlaced processes. Initially, assume that an available argument A of Ag is acceptable w.r.t. a set of arguments S if every argument against A from $\mathcal{A}rgue_{Id(Ag)}$ is attacked by an argument in S. Intuitively, if an agent Ag builds an available argument A by concatenating its local partial argument with arguments from $\mathcal{C}ooperate_{Id(Ag)}$,

[2] $(1, [b;\ a \leftarrow b])$ can also be obtained from the set of available arguments

$$\{(1, [a \leftarrow b]), (2, [b])\}$$

Both ways to complete the local partial argument $(1, [a \leftarrow b])$ of agent Ag_1 are correct, but the former is less intuitive. So, we prefer to illustrate it in the former way.

then A must be evaluated by every argumentative agent in $\mathcal{A}rgue_{Id(Ag)}$. It means that each argumentative agent Ag^a should try to build an available argument CA against A. Two situations may occur:

1. Ag^a argues and cooperates only with itself. If Ag^a cannot build a complete argument CA by itself, and since there is no other agent to cooperate with Ag^a, Ag^a cannot argue against A. On the other hand, if CA is built by Ag^a, Ag^a does not need evaluation of CA by any other agent than itself, and so Ag^a might use its argument against A; or
2. Ag^a argues and/or cooperates with other agents. In such a case, to build a CA may require the concatenation of arguments from $\mathcal{C}ooperate_{Id(Ag^a)}$ and then the evaluation of CA by agents in $\mathcal{A}rgue_{Id(Ag^a)}$. The argumentative process for CA of Ag^a finishes when the acceptability of CA with respect to arguments from $\mathcal{A}rgue_{Id(Ag^a)}$ is obtained.

Independently of which situation occurs for each $Ag^a \in \mathcal{A}rgue_{Id(Ag)}$, if there exists at least one acceptable argument CA from $\mathcal{A}rgue_{Id(Ag)}$ against the available argument A of Ag, then A is not acceptable (with respect to $\mathcal{A}rgue_{Id(Ag)}$); otherwise, A is an acceptable argument.

The following example illustrates the above informal description.

Example 2. Let be $\mathcal{A} = \{Ag_1, Ag_2, Ag_3, Ag_4, Ag_5, Ag_6\}$ such that each agent is

$$\begin{array}{rcl} Ag_1 & = & < 1, \{a \leftarrow not\ b, c;\ c\},\ \{1,2,3\},\ \{1\}\ > \\ Ag_2 & = & < 2, \{b \leftarrow not\ d, f\},\ \{2,5\}, \{2,4\}\ > \\ Ag_3 & = & < 3, \{b \leftarrow d\},\ \{3\}, \{3\}\ > \\ Ag_4 & = & < 4, \{f\},\ \{4\}, \{4\}\ > \\ Ag_5 & = & < 5, \{d \leftarrow not\ g\},\ \{5,6\}, \{5\}\ > \\ Ag_6 & = & < 6, \{g\},\ \{6\}, \{6\}\ > \end{array}$$

Assume that Ag_1 needs to deduce the acceptability of an argument for a. These are the steps to the solution: Ag_1 should have an argument for the objective literal a and such an argument must be acceptable w.r.t. $\mathcal{A}rgue_1 = \{2, 3\}$. Since Ag_1 has an argument $A_1^s(a) = (1, [c; a \leftarrow not\ b, c])$, Ag_2 and Ag_3 should have an argument against $A_1^s(a)$.

1. Ag_3 does not have any argument against $A_1^s(a)$ because it has only a partial argument for b and there is no argument from $\mathcal{C}ooperate_3$ to complete it.
2. Ag_2 has a partial argument for b that can be completed by Ag_4's argument for f, i.e. $A_2^s(b) = (2, [f; b \leftarrow not\ d, f]) \in (2, [b \leftarrow not\ d, f]) + (4, [f])$. So, Ag_5 should have an argument against $A_2^s(b)$.
   (a) Ag_5 has the argument $A_5^s(d) = (5, [d \leftarrow not\ g])$, and now agent Ag_6 should have an argument against $A_5^s(d)$.
      i. Ag_6 has the argument $A_6^s(g) = (6, [g])$. The argument $A_6^s(g)$ is, therefore, acceptable because there is no argument from $\mathcal{A}rgue_6$ against it.

   Thus, $A_5^s(d)$ is not acceptable because it is attacked by $A_6^s(g)$.

   Since Ag_5 has no acceptable argument against $A_2^s(b)$, $A_2^s(b)$ is an acceptable argument w.r.t. arguments from $\mathcal{A}rgue_2$.

Finally, $A_1^s(a)$ is not acceptable because there is at least one acceptable argument from $\mathcal{A}rgue_1$ against it, viz. $A_2^s(b)$.

We proceed by exposing the required definitions for this informal description. First of all, it is necessary to determine the available arguments that can be used to attack. As expected, only complete arguments in $\mathcal{A}rgs(\mathcal{A})$ should be considered. These arguments are called *attacking arguments*.

Definition 11 (Attacking Argument). *Let $\mathcal{A}$ be a MAS, α an agent's identity in $\mathcal{A}$, and $S \subseteq \mathcal{A}rgs(\mathcal{A})$. (α, Seq) is an* attacking argument *given S iff it is a complete argument and belongs to $Av(S)$. If (α, Seq) is either a s-argument or a w-argument, we refer to it by s-attacking or w-attacking argument, respectively.*

Intuitively, both strong and weak arguments can be attacked in the same way. Since a (weak or strong) argument makes assumptions, other arguments for the complement of one such assumption may attack it. In other words, an argument with $not\ L$ can be attacked by arguments for L. This definition of attack encompasses the case of arguments that are directly conflicting, e.g. an argument for L (with $not\ \neg L$) can be attacked by an argument for $\neg L$. The previous claim that any weak argument $A_\alpha^w(L) = (\alpha, Seq_L^w)$ (and also a strong argument $A_\alpha^s(L) = (\alpha, Seq_L^s)$ which verifies $not\ \bot \in Assump(Seq_L^s)$) can be attacked by every argument for $\bot$. However, it does not make sense to attack arguments for objective literals if they do not lead to *falsity*. By "an objective literal L leads to *falsity*" we mean that there is an argument $A_\alpha(L)$ such that $A_\beta(\bot)$ is built based on such an argument, e.g.

$$A_\beta^s(\bot) : A_\alpha^s(L) + [\bot \leftarrow L, not\ L']$$

We only consider objective literals that are in the body of the rule for $\bot$ because these literals immediately lead to *falsity*. We assume that the involvement of other objective literals are not as strong as those in the body of the denial[3]. Then objective literals are *directly conflicting with* $A_\beta(\bot)$ if the following holds:

Definition 12 (Directly Conflict with $A_\bot$). *Let $A_\beta(\bot)$ be an argument of β for $\bot$, '$\bot \leftarrow Body$' be the rule in $A_\beta(\bot)$ and $\{L_1, \ldots, L_n\}$ be the set of all objective literals in $Body$. The set of objective literals* directly conflicting with $A_\beta(bot)$ *is*

$$DC(Seq_\bot) = \{\bot\} \cup \{L_1, \ldots, L_n\}$$

If an argument of α for L has a default negation $not\ L'$ in it, any argument for L' attacks $A_\alpha(L)$ (by undercut [11]). The other attacking relation (called rebut [11]) states that an argument also attacks another one when both arguments have complementary conclusions (i.e. one concludes L and the other $\neg L$). With strong and weak arguments, *rebut can be reduced to undercut*[4]. So, we can say informally that "an argument of α

[3] We further assume they can be detected in a process of "belief revision", e.g. [4]. However, a discussion of this issue is beyond the scope of this proposal.

[4] This simplification has been proposed in [3,5,14]. [3] defines a methodology for transforming non-exact, defensible rules into exact rules with explicit non-provability conditions and shows that this transformation eliminates the need for rebuttal attacks and for dealing with priorities in the semantics. In [14,5], it is proposed that "attacks" can be reduced to "undercut" by considering weaker version of arguments.

for a conclusion L *attacks* an argument of β with an assumption $not\ L$". Such a "notion of attack" shows that we need to make both the conclusions and the assumptions of an argument precise before defining an attack.

Definition 13 (Conclusions and Assumptions). *Let $A_\alpha(L) = (\alpha, Seq_L)$ be an argument of α for L. The* conclusions *of $A_\alpha(L)$, $Conc(Seq_L)$, is the set of all objective literals that appear in the head of rules in $A_\alpha(L)$. The* assumptions *of $A_\alpha(L)$, $Assump(Seq_L)$, is the set of all default literals appearing in the bodies of rules in $A_\alpha(L)$.*

Intuitively, we want to define attack in terms of both attacking and available arguments. However, we still need to determine which attacking arguments can be used to attack available ones. Moreover, to prevent cyclic definitions, an attack is defined only in terms of arguments.

Definition 14 (Attack). *Let $\mathcal{A}$ be a MAS, α and β be agent's identifiers in $\mathcal{A}$, and $\mathcal{A}rgue_\alpha$ be the α's set of argumentative agents. An argument (β, Seq_{L_1}) of β for L_1* attacks *an argument (α, Seq_{L_2}) of α for L_2 iff*

- $\beta \in \mathcal{A}rgue_\alpha$*; and*
- Seq_{L_1} *is a well-defined sequence over R_β, or $\alpha \in \mathcal{A}rgue_\beta$ and*

$$Seq_{L_1} \in Seq_{L_2} + Seq'_{L_1}$$

  where Seq'_{L_1} is a well-defined sequence for L_1 over R_β; and
- L_1 *is the symbol $\bot$, $not\ \bot \in Assump(Seq_{L_2})$ and $L_2 \in DC(Seq_{L_1})$, or L_1 is an objective literal different from $\bot$ and $not\ L_1 \in Assump(Seq_{L_2})$.*

Recall that, as with other argumentation based frameworks the semantics is defined based on a notion of acceptable arguments, where a set of arguments is acceptable if any argument attacking it is itself attacked by the set. Now, in this distributed setting, care must be taken about which arguments can be used to attack a set of arguments, and which arguments are available for being used to defend the attacks. Before presenting the definition of acceptable arguments we motivate for the definition in such a distributed setting. Moreover, note that the above definition of attack has a condition that foresees cases where "indirect cooperation" between argumentative agents is needed. The following example illustrates such a situation.

Example 3. Consider $\mathcal{A} = \{Ag_1, Ag_2\}$ where:

$$\begin{array}{rcl} Ag_1 & = & < 1, \{c; a \leftarrow c, not\ b\},\ \{1,2\},\ \{1\} > \\ Ag_2 & = & < 2, \{b \leftarrow c; z \leftarrow not\ a\},\ \{1,2\},\ \{2\} > \end{array}$$

The set of available arguments of $\mathcal{A}$ given $LA(\mathcal{A})$ is

$$\mathcal{A}rgs(\mathcal{A}) = \{A_1^s(c), A_1^s(a), PA_2^s(b), A_2^s(z)\}$$

Moreover, from Definition 14, the complete argument $A_1^s(a)$ attacks $A_2^s(z)$ and the partial argument $PA_2^s(b)$ attacks $A_1^s(a)$. However, we only want attacking arguments

(i.e. complete arguments) to be used to determine the acceptability of an argument w.r.t. $\mathcal{A}rgs(\mathcal{A})$. Then, $PA_2^s(b)$ will not be used and, consequently, $A_2^s(z)$ is not acceptable. Nevertheless, $A_1^s(a)$ has a rule for c that can be used to complete $PA_2^s(b)$. If agent Ag_2 may 'use' such a rule from $A_1^s(a)$ to complete its partial argument $PA_2^s(b)$, Ag_2 has an argument

$$(2, [c; b \leftarrow c])$$

that can be used against $A_1^s(a)$. Therefore, $A_2^s(z)$ is acceptable w.r.t. $\mathcal{A}rgs(\mathcal{A})$.

At this point, it is quite clear that we should evaluate available arguments of a multi-agent setting $\mathcal{A}$ to conclude which of them are acceptable with respect to a set S of arguments (that are already considered acceptable with respect to a set of arguments from $\mathcal{A}$). However, should an argument of an agent Ag be acceptable in $\mathcal{A}rgue_{Ag}$ if such an argument is to be used in a cooperation process? For instance, consider:

$$Ag_1 = < 1, \{q \leftarrow a; c\}, \{1\}, \{1, 2\} >$$
$$Ag_2 = < 2, \{a \leftarrow not\ b; b \leftarrow not\ a, not\ c\}, \{2\}, \{2\} >$$

and assume that every argument in $LA(\mathcal{A})$ is a strong argument.

For having an acceptable argument for q in Ag_1, Ag_1 must complete its available argument for q, viz. $PA_1^s(q) = (1, [q \leftarrow a])$. Agent Ag_2 has an available argument for a, $A_2^s(a) = (2, [a \leftarrow not\ b])$. However, Ag_2 has also an attacking argument $A_2^s(b) = (2, [b \leftarrow not\ a, not\ c])$ against $A_2^s(a)$. Two possible approaches can deal with this situation:

1. since both arguments attack each other, neither $A_2^s(a)$ nor $A_2^s(b)$ are acceptable in $\mathcal{A}rgue_2$, and so $A_2^s(a)$ cannot be used to complete $PA_1^s(q)$; or
2. since there is no acceptable argument in $\mathcal{A}rgue_2$ attacking $A_1^s(a)$, it is defensible. Furthermore, $A_1^s(a)$ is used to complete $PA_1^s(q)$ and so the resulting available argument is

   $$A_1^s(q) = [a \leftarrow not\ b; q \leftarrow a]$$

   However, $A_1^s(q)$ should be evaluated by $\mathcal{A}rgue_1$. Via cooperation, Ag_1 has an attacking argument

   $$A_1^s(b) = (1, [b \leftarrow not\ a, not\ c])$$

   against $A_1^s(q)$. But Ag_1 has also an attacking argument $A_1^s(c) = (1, [c])$ against $A_1^s(b)$ which no argument attacks. Thus, $A_1^s(c) = (1, [c])$ is acceptable and, consequently, $A_1^s(b)$ is not acceptable (both with respect to arguments from $\mathcal{A}rgue_1$). Therefore, $A_1^s(q)$ is acceptable with respect to arguments from $\mathcal{A}rgue_1$.

The second approach allows us to draw more acceptable arguments than the first one. In fact, the arguments evaluated are acceptable if we consider the overall agent's knowledge. Moreover, this approach is more credulous than the first one. Therefore, we follow the latter and define that for a given agent Ag in a multi-agent setting $\mathcal{A}$, an agent $Ag^c \in \mathcal{C}ooperate_{Id(Ag)}$ cooperates with an available argument A under one of the following conditions: (i) A is not attacked by any argument from $\mathcal{A}rgue_{Id(Ag^c)}$, or (ii) A is attacked, but every attacking argument B against A is attacked by some argument from $\mathcal{A}rgue_{Id(Ag^c)}$. In both cases, A is considered a *defensible argument*. The

following operator defines which are the defensible arguments, given a set of available arguments of a multi-agent setting. In the remainder, we use the notation p and o to distinguish the proposed argument from the opposing one, i.e. p (resp. o) is a (strong or weak) proposed (resp. opposing) argument.

Definition 15 (Operator $Def_{p,o}(S)$). *Let $\mathcal{A}$ be a MAS, $\mathcal{A}rgs(\mathcal{A})$ be the set of available arguments of $\mathcal{A}$, $S \subseteq \mathcal{A}rgs(\mathcal{A})$ be a set of p-arguments. $Def_{p,o}(S)$ is the set of all o-arguments of $\mathcal{A}rgs(\mathcal{A})$ that are not attacked by any attacking argument given S. Arguments in $Def_{p,o}(S)$ are called* defensible arguments*.*

Assume that arguments in the set of defensible arguments are opposing arguments, and every argument in a set of available arguments is a proposed argument. Now we can determine how available p-arguments are acceptable with respect to a set S of p-arguments from $\mathcal{A}rgs(\mathcal{A})$, such that S is a pre-defined set of acceptable arguments. In order to determine the set of acceptable arguments with respect to S, the following steps must be performed:

1. obtain the opposing arguments via $Def = Def_{p,o}(S)$. This encompasses the following two sub-steps:
   (a) get the set $Atts$ of p-attacking arguments given S, i.e. the complete arguments in $Av(S)$. This sub-step also allows p-partial arguments in S to be completed by arguments in S and so new p-arguments are built;
   (b) reject o-arguments in $\mathcal{A}rgs(\mathcal{A})$ that are attacked by arguments in $Atts$.
2. obtain the proposed (partial or complete) arguments given S, i.e. $Av(S)$. This sub-step also allows p-partial arguments to be completed by arguments in S and so new p-arguments are built.
3. determine which are (i) the opposing arguments attacking some proposed argument and (ii) the opposing arguments attacked by arguments in S.

Definition 16 (Acceptable Argument). *Let $\mathcal{A}$ be a MAS, $\mathcal{A}rgs(\mathcal{A})$ be the set of arguments of $\mathcal{A}$, $S \subseteq \mathcal{A}rgs(\mathcal{A})$ be a set of p-arguments, and α, β, and γ be agent's identifiers in $\mathcal{A}$. A p-argument (α, Seq_L) for a literal L is $acceptable_{p,o}$* w.r.t. *S iff (i) it is either a local argument, or it belongs to the set of available arguments given S; and (ii) for any o-attacking argument $(\beta, Seq_{L'})$ for a literal L' given $Def_{p,o}(S)$: if $(\beta, Seq_{L'})$ attacks (α, Seq_L) then there exists a complete p-argument $(\gamma, Seq_{L''})$ for a literal L'' in S that attacks $(\beta, Seq_{L'})$.*

We now formalise the concept of acceptable arguments with a fixpoint theory and also define a characteristic function $p\ o$ of multi-agent setting $\mathcal{A}$ over a set of acceptable arguments S as follows:

Definition 17 (Characteristic Function). *Let $\mathcal{A}$ be a MAS, $\mathcal{A}rgs(\mathcal{A})$ be the set of available arguments of $\mathcal{A}$, and $S \subseteq \mathcal{A}rgs(\mathcal{A})$ be a set of p-arguments. The* characteristic function *$p\ o$ of $\mathcal{A}$ over S is:*

$$F_{\mathcal{A}}^{p,o} : 2^{\mathcal{A}rgs(\mathcal{A})} \rightarrow 2^{\mathcal{A}rgs(\mathcal{A})},$$
$$F_{\mathcal{A}}^{p,o}(S) = \{Arg \in \mathcal{A}rgs(\mathcal{A}) \mid A\ is\ acceptable_{p,o}\ w.r.t.\ S\}$$

We can see that, if an argument A is $acceptable_{p,o}$ w.r.t. S, A is also $acceptable_{p,o}$ w.r.t. any superset of S. In fact, it can be shown that $Def_{p,o}(S)$ is anti-monotonic, and so $F_{\mathcal{A}}^{p,o}$ is monotonic. Being monotonic, it is guaranteed that $F_{\mathcal{A}}^{p,o}$ always has a least fixpoint (according to the set inclusion ordering over sets of arguments):

Proposition 3. *Define for any $\mathcal{A}$ the following transfinite sequence of sets of arguments:*

$$\begin{array}{ll} S^0 & = \emptyset \\ S^{i+1} & = F_{\mathcal{A}}^{p,o}(S^i) \\ S^\delta & = \bigcup_{\alpha<\delta} S^\alpha \textit{ for limit ordinal } \delta \end{array}$$

1. *$F_{\mathcal{A}}^{p,o}$ is monotonic, and so there must exist a smallest λ such that S^λ is a fixpoint of $F_{\mathcal{A}}^{p,o}$, and $S^\lambda = lfp(F_{\mathcal{A}}^{p,o})$.*
2. *If $F_{\mathcal{A}}^{p,o}$ is finitary then $lfp(F_{\mathcal{A}}^{p,o}) = F_{\mathcal{A}}^{p,o\uparrow\omega}(\emptyset)$.*

By knowing the set S of all acceptable arguments of $\mathcal{A}$, we can split all complete arguments from $\mathcal{A}rgs(\mathcal{A})$ into three classes: justified arguments, overruled arguments or defensible arguments. An argument A is *justified* when A is in S. An argument A is *overruled* when A is attacked by at least one argument in S. Finally, an argument A is *defensible* when A is attacked by an argument $B \in \mathcal{A}rgs(\mathcal{A})$, and neither A nor B are attacked by acceptable arguments.

Definition 18 (Justified, Overruled or Defensible Argument). *Let $\mathcal{A}$ be a MAS, $\mathcal{A}rgs(\mathcal{A})$ be the set of available arguments of $\mathcal{A}$, $S \subseteq \mathcal{A}rgs(\mathcal{A})$, and $F_{\mathcal{A}}^{p,o}$ be the characteristic function p o of $\mathcal{A}$ and over S. A complete p-argument for a literal L of an agent with identity α is:*

- justified$_{\mathcal{A}}^{p,o}$ *iff it is in $lfp(F_{\mathcal{A}}^{p,o})$*
- overruled$_{\mathcal{A}}^{p,o}$ *iff there exists a justified$_{\mathcal{A}}^{o,p}$ o-argument for a literal L' of an agent β in $\mathcal{A}$ attacking it*
- defensible$_{\mathcal{A}}^{p,o}$ *iff it is neither justified$_{\mathcal{A}}^{p,o}$ nor overruled$_{\mathcal{A}}^{p,o}$.*

We denote the $lfp(F_{\mathcal{A}}^{p,o})$ by $JustArgs_{\mathcal{A}}^{p,o}$.

Example 4. Let $\mathcal{A} = \{Ag_1, Ag_2, Ag_3\}$ such that each agent is

$$\begin{array}{rcl} Ag_1 & = & < 1, \{\ a \leftarrow not\ b\ \}, \{1,2\},\ \{1\}\} > \\ Ag_2 & = & < 2, \{\ b \leftarrow not\ c\}, \{2,3\},\ \{2\}\ > \\ Ag_3 & = & < 3, \{\ c \leftarrow not\ a\ \}, \{2,3\},\ \{3\}\} > \end{array}$$

In this example we show how to obtain $lfp(F_{\mathcal{A}}^{s,s}(\emptyset))$. First of all, we determine the set of strong local arguments of $\mathcal{A}$:

$$LA^s(\mathcal{A}) = \left\{ \begin{array}{ll} (1,[]), & (1,[a \leftarrow not\ b]), \\ (2,[]), & (2,[b \leftarrow not\ c]), \\ (3,[]), & (3,[c \leftarrow not\ a]) \end{array} \right\}$$

and the set of available arguments of $\mathcal{A}$ given $LA^s(\mathcal{A})$, i.e $\mathcal{A}rgs(\mathcal{A}) = LA^s(\mathcal{A})$

– let $S^0 = \emptyset$. Since $Atts^0 = \emptyset$, the set of opposing arguments is

$$Def^0 = Def_{s,s}(S^0) = \{(1, [a \leftarrow not\ b]), (2, [b \leftarrow not\ c]), (3, [c \leftarrow not\ a])\}$$

The set of proposed arguments is $Av(S^0) = LA^s(\mathcal{A})$. Then we determine the following attacks

opposing argument	*proposed argument*
$(2, [b \leftarrow not\ c])$	$(1, [a \leftarrow not\ b])$
$(3, [c \leftarrow not\ a])$	$(2, [b \leftarrow not\ c])$
	$(3, [c \leftarrow not\ a])$
	$(1, []), (2, []), (3, [])$

So $S^1 = F^{s,s}_{\mathcal{A}}(S^0) = \{(3, [c \leftarrow not\ a]), (1, []), (2, []), (3, [])\}$;

– since $Atts^1 = \{(3, [c \leftarrow not\ a])\}$,

$$Def^1 = Def_{s,s}(S^1) = \{(1, [a \leftarrow not\ b]), (3, [c \leftarrow not\ a])\}$$

The set of proposed arguments is $Av(S^1) = S^1$. Despite the opposing argument $A^s_2(b) = (2, [b \leftarrow not\ c])$ attacks the proposed argument $(1, [a \leftarrow not\ b])$, $A^s_2(b)$ is attacked by the acceptable argument $(3, [c \leftarrow not\ a])$, so

$$S^2 = F^{s,s}_{\mathcal{A}}(S^1) = S^1 \cup \{(1, [a \leftarrow not\ b])\}$$

– since $F^{s,s}_{\mathcal{A}}(S^2) = S^2$, the set of justified$^{s,s}_{\mathcal{A}}$ arguments is

$$JustArgs^{s,s}_{\mathcal{A}} = \{(3, [c \leftarrow not\ a]), (1, [a \leftarrow not\ b]), (1, []), (2, []), (3, [])\}.$$

Argument $(2, [b \leftarrow not\ c])$ is overruled$^{s,s}_{\mathcal{A}}$ because it is attacked by the justified$^{s,s}_{\mathcal{A}}$ argument $(3, [c \leftarrow not\ a])$. No argument in $\mathcal{A}rgs(\mathcal{A})$ is defensible$^{s,s}_{\mathcal{A}}$.

3 Properties

Here we assume very little about the sets of argumentative and cooperative agents of an agent. By imposing restriction on these sets different properties of the whole setting can be obtained. In particular, as expected, if all agents in a multi-agent setting $\mathcal{A}$ argue and cooperate with all others, then the result is exactly the same as having a single agents with the whole knowledge:

Theorem 1. *Let $\mathcal{A}$ be a MAS over $\mathcal{L}$ and Ids such that for every agent $Ag \in \mathcal{A}$: $\mathcal{C}ooperate_{Id(Ag)} = Ids$ and $\mathcal{A}rgue_{Id(Ag)} = Ids$, and $F^{p,o}_{\mathcal{A}}$ be the characteristic function p o of $\mathcal{A}$. Let $P = \{< \beta, Kb_\beta, \{\beta\}, \{\beta\} >\}$ such that*

$$Kb_\beta = \bigcup_{\alpha_i \in Ids} Kb_{\alpha_i}$$

and $F^{p,o}_P$ be the characteristic function p o of P.

Then, for every agent $\alpha_i \in Ids$: $(\alpha_i, Seq) \in lfp(F^{p,o}_{\mathcal{A}})$ iff $(\beta, Seq) \in lfp(F^{p,o}_P)$.

Corollary 1. *If $\mathcal{A}$ is as in Theorem 1 then for any pair of agents in $\mathcal{A}$, with identifiers α_i and α_j :, $(\alpha_i, Seq) \in lfp(F_{\mathcal{A}}^{p,o})$ iff $(\alpha_j, Seq) \in lfp(F_{\mathcal{A}}^{p,o})$.*

However, the semantics at one agent can different from that of the union, as desired:

Example 5. Consider $\mathcal{A} = \{Ag_1, Ag_2, Ag_3\}$ such that each agent is

$$\begin{array}{rcl} Ag_1 & = & < 1, \{\ a \leftarrow not\ b\ \},\ \{1,2,3\},\ \{1,2,3\}\} > \\ Ag_2 & = & < 2, \{\ b \leftarrow not\ c\},\ \{1,2,3\},\ \{1,2,3\}\ > \\ Ag_3 & = & < 3, \{\ c \leftarrow not\ a\ \},\ \{1,2,3\},\ \{1,2,3\}\} > \end{array}$$

$Def_{s,s}(\emptyset) = \{(1,[]),(2,[]),(3,[]),(1,[a \leftarrow not\ b]),(2,[a \leftarrow not\ b]),(3,[a \leftarrow not\ b]),(1,[b \leftarrow not\ c]),(2,[b \leftarrow not\ c]),(3,[b \leftarrow not\ c]),(1,[c \leftarrow not\ a]),(2,[c \leftarrow not\ a]),(3,[c \leftarrow not\ a])\}$. The arguments $A^s(a)$, $A^s(b)$, and $A^s(c)$) are attacked[5]. As there is no "counter-attack" to any of those attacks, $lfp(F_{\mathcal{A}}^{s,s}(\emptyset)) = JustArgs_{\mathcal{A}}^{s,s} = \emptyset$. No argument in $\mathcal{A}rgs(\mathcal{A})$ is overruled$_{\mathcal{A}}^{s,s}$, and all of arguments are concluded to be defensible$_{\mathcal{A}}^{s,s}$. However, we obtain a different result if we consider that

$$\begin{array}{rcl} Ag_1 & = & < 1, \{\ a \leftarrow not\ b\ \},\ \{1,2\},\ \{1,2\}\} > \\ Ag_2 & = & < 2, \{\ b \leftarrow not\ c\},\ \{2,3\},\ \{2,3\}\ > \\ Ag_3 & = & < 3, \{\ c \leftarrow not\ a\ \},\ \{3\},\ \{3\}\} > \end{array}$$

Here $Def_{s,s}(\emptyset) = \{(1,[]),(2,[]),(3,[]),(1,[a \leftarrow not\ b]),(1,[b \leftarrow not\ c]),(1,[c \leftarrow not\ a]),(2,[b \leftarrow not\ c]),(2,[c \leftarrow not\ a]),(3,[c \leftarrow not\ a])\}$, and $lfp(F_{\mathcal{A}}^{s,s}(\emptyset)) = JustArgs_{\mathcal{A}}^{s,s} = \{A_2^s(c), A_3^s(c)\}$. Note here how the result differs also from agent to agent.

Due to space limitations we do not detail here general properties when some other weaker restriction are imposed (e.g. imposing transitivity, symmetry in the cooperation or argumentation set, etc). Instead, we discuss about some properties of $JustArgs_{\mathcal{A}}^{p,o}$ and comparisons. Since p (resp. o) denote the kind of a proposed (resp. an opposing) argument, i.e. strong argument or weak argument, assume that p (resp. o) in $\{s, w\}$. Both $JustArgs_{\mathcal{A}}^{w,w}$ and $JustArgs_{\mathcal{A}}^{w,s}$ are both conflict-free[6] and non-contradictory[7]. Thus, every argument in both $JustArgs_{\mathcal{A}}^{w,w}$ and $JustArgs_{\mathcal{A}}^{w,s}$ is non-contradictory, i.e. it is not related to a contradiction at all. Furthermore, $F_{\mathcal{A}}^{w,w}$ has more defensible arguments than $F_{\mathcal{A}}^{w,s}$. Therefore, we obtain a consistent way of reasoning in a multi-agent setting $\mathcal{A}$ if we apply $F_{\mathcal{A}}^{w,w}$ over $\mathcal{A}rgs(\mathcal{A})$.

In contrast, $JustArgs_{\mathcal{A}}^{s,s}$ and $JustArgs_{\mathcal{A}}^{s,w}$ may be contradictory. However, to evaluate the acceptability of available arguments without considering the presence of *falsity* or both arguments for L and $\neg L$, the proposed arguments should be strong ones, and every opposing argument is a weak argument. Since $F_{\mathcal{A}}^{s,w}$ respects the 'Coherence

[5] For simplicity, since every agent argues with every other, we omit agent identity of the arguments.

[6] A set S of arguments is conflict-free if there is no argument in S attacking an argument in S.

[7] A set S of arguments is non-contradictory if neither an argument for *falsity* nor both arguments for L and $\neg L$ are in S.

Principle' of [9,1], i.e. given that every opposing argument is a weak one, it can be attacked by any proposed argument for its explicit negation. Therefore, we obtain a paraconsistent way of reasoning in a multi-agent setting $\mathcal{A}$ if we apply $F_{\mathcal{A}}^{s,w}$ over $Args(\mathcal{A})$. Moreover, a justified$_{\mathcal{A}}^{s,w}$ argument of an agent in $\mathcal{A}$ is related to a contradiction with respect to $JustArgs_{\mathcal{A}}^{s,w}$:

Definition 19 (Relation to a Contradiction). *Let $\mathcal{A}$ be a MAS, α and β be agents' identity in MAS, $\beta \in Argue_{\alpha}$, and $JustArgs_{\mathcal{A}}^{s,w}$ be the $lfp(F_{\mathcal{A}}^{s,w})$. A justified$_{\mathcal{A}}^{s,w}$ s-argument $A_{\alpha}^{s}(L) = (\alpha, Seq_L)$ is:*

- *contradictory$_{\mathcal{A}}^{s,w}$ if L is the symbol $\bot$, or there exists a justified$_{\mathcal{A}}^{s,w}$ s-argument $(\beta, Seq_{\bot})$ such that $L \in DC(Seq_{\bot})$, or there exists a justified$_{\mathcal{A}}^{s,w}$ s-argument $(\beta, Seq_{\neg L})$; or*
- *based-on-contradiction$_{\mathcal{A}}^{s,w}$ if $A_{\alpha}^{s}(L)$ is justified$_{\mathcal{A}}^{s,w}$, it does not exists a justified$_{\mathcal{A}}^{s,w}$ s-argument $(\beta, Seq_{\neg L})$ and $A_{\alpha}^{s}(L)$ is also overruled$_{\mathcal{A}}^{s,w}$; or*
- *non-contradictory$_{\mathcal{A}}^{s,w}$, otherwise.*

As already said, any agent's belief should be concluded only with respect to both sets of argumentative and cooperative agents with such an agent. Intuitively, we can conclude that different truth values of a given literal L over a multi-agent setting $\mathcal{A}$ might be obtained. It happens because it depends on which agent the literal L is inferred from, and also on what the specification of both sets of cooperative and argumentative agents is, given the overall agents in $\mathcal{A}$. Then, a truth value of an agent's conclusion in a (consistent or paraconsistent) way of reasoning is as follows:

Definition 20 (Truth Value of an Agent's Conclusion). *Let $\mathcal{A}$ be a MAS, α is an agent's identity of $\mathcal{A}$, $k \in \{s, w\}$, and L be an objective literal or the symbol $\bot$. L over $\mathcal{A}$ is:*

- false$_{\alpha}^{k,w}$ *iff for all argument of α for L: it is overruled$_{\mathcal{A}}^{k,w}$*
- true$_{\alpha}^{k,w}$ *iff there exists a justified$_{\mathcal{A}}^{k,w}$ argument of α for L. Moreover, L is*
  - contradictory$_{\alpha}^{k,w}$ *if L is the symbol $\bot$ or there exists a justified$_{\mathcal{A}}^{k,w}$ argument of α for $\neg L$*
  - based-on-contradiction$_{\alpha}^{k,w}$ *if it is both true$_{\alpha}^{k,w}$ and false$_{\alpha}^{k,w}$*
  - non-contradictory$_{\alpha}^{k,w}$, *otherwise.*
- undefined$_{\alpha}^{k,w}$ *iff L is neither true$_{\alpha}^{k,w}$ nor false$_{\alpha}^{k,w}$.*

Note that this point that truth is defined parametric of the agent. So, it is only natural that the truth value of a proposition may differ from agent to agent.

Proposition 4. *Let $k \in \{s, w\}$. L is undefined$_{\alpha}^{k,w}$ iff there is no justified$_{\mathcal{A}}^{k,w}$ argument of α for L and at least one argument of α for L is not overruled$_{\mathcal{A}}^{k,w}$.*

This paraconsistent semantics for multiple logic programs is in accordance with the paraconsistent well-founded semantics $WFSX_p$ [1]. In fact, both coincide if there is a single program (or a set, in case all cooperate and argue with all other, cf. Theorem 1):

Theorem 2 ($WFSX_p$ semantics vs $F_{\mathcal{A}}^{s,w}$). *Let P be an ELP such that $\bot \notin \mathcal{H}(P)$, and let L be an objective literal in $\mathcal{H}(P)$. $L \in WFSX_p(P)$ iff L is true$_{\mathcal{A}}^{s,w}$, not $L \in WFSX_p(P)$ iff L is false$_{\mathcal{A}}^{s,w}$, and $\{L, not\ L\} \cap WFSX_p(P) = \emptyset$ iff L is undefined$_{\mathcal{A}}^{s,w}$.*

Moreover, there is a relation between the consistent reasoning is obtained of $F_{\mathcal{A}}^{w,w}$ and [6]'s grounded (skeptical) extension if the following holds. To show this, we first relate [6]'s definitions of both *RAA-attack* and *g-attack* to our definition of *attack* as follows:

Lemma 1. *Let P be an ELP such that $\bot \notin \mathcal{H}(P)$, (A_L, L) be an argument for L, $\{(A_L, L), (A_{L'}, L'), (A_{\neg L}, \neg L)\} \subseteq Args(P)$ such that not $L \in A_{L'}$, $\{1, 2, 3\}$ is a subset of agent's identities of $\mathcal{A}$, and $\{(1, Seq_L^w), (2, Seq_{L'}^w), (3, Seq_{\neg L}^w)\} \in \mathcal{A}rgs(\mathcal{A})$ such that not $L \in Assump(Seq_{L'}^w)$.*

If (A_L, L) g-attacks *$(A_{L'}, L')$ then $(1, Seq_L^w)$* attacks *$(2, Seq_{L'}^w)$.*

If $(A_{\neg L}, \neg L)$ RAA-attacks *(A_L, L) then $(3, Seq_{\neg L}^w)$* attacks *$(1, Seq_L^w)$.*

Theorem 3 (Grounded extension vs $F_{\mathcal{A}}^{w,w}$). *Let P be an ELP such that $\bot \notin \mathcal{H}(P)$, L be an objective literal in $\mathcal{H}(P)$, B be the Ground Extension's characteristic function of P, (A_L, L) be an argument for L, and α be an agent's identity of $\mathcal{A}$.*

An argument $(A_L, L) \in lfp(B)$ iff $\exists(\alpha, Seq_L^w) \in lfp(F_{\mathcal{A}}^{w,w})$.

An argument $(\{not\ L\}, L) \in lfp(B)$ iff $\neg\exists(\alpha, Seq_L^w) \in gfp(F_{\mathcal{A}}^{w,w})$.

4 Illustrative Examples

To illustrate the results of the proposed semantics, we present here some examples. The first example illustrates how the framework can be used to model over a hierarchy and in the second to model evolving knowledge.

Example 6 (Business Process Management). This example is derived from ADEPT project[8] which developed negotiating agents for business process management. One such process deals with the provision of customer quotes for networks adapted to the customer's needs. The agents' interaction involves both argumentation and cooperation as follows: *customer service division* (CSD) must not quote if the customer is not credit-worthy which it should assume by default.

So, CSD should obtain an agreement with *vet customer* (VC) which means that VC may counter-argue and give evidence for the credit-worthiness of the customer. In case credit is approved, if CSD does not have a portfolio item for the solicited quote, it needs from *design department* (DD) a quote for it. DD might do this task if *surveyor department* (SD) does not argue that such a task is not important. DD needs information held by CSD.

Considering first the *customer service division*, it knows about the client's equipment (dubbed eq) and its requirements (dubbed req): requirements 2 and 3, and equipments 2 and 3. Furthermore, CSD knows the customer is important. These can be represented as facts: $req(2)$; $req(3)$; $eq(2)$; $eq(3)$; and $important$. Besides these facts about a particular client, CSD has general rules such as requirements 1, 2 and 3 together make up a portfolio and can be quoted if a previous quote exists (otherwise, the DD has to prepare a quote):

$$portofolio \leftarrow req(1), req(2), req(3)$$
$$quote \leftarrow portfolio, previousQuote$$

[8] See details in *http://lsdis.cs.uga.edu/Projects/*

CSD does not provide a quote if the client is not credit-worthy:

$$\neg quote \leftarrow not\ creditWorthy$$

The *Vet Customer* knows the client is not credit-worthy: it has a fact $\neg creditWorthy$. The *design department* knows that there is no need to survey the client site if the client has equipments 1, 2 and 3. It can be represented by the rule:

$$\neg need2survey \leftarrow eq(1), eq(2), eq(3)$$

In general, DD assumes that the SD does a survey unless it is busy which can be represented by the rule $survey \leftarrow not\ busySD$. The quote of DD can be obtained by a simple design cost if there was no need survey; otherwise, by a complex design cost:

$$quote \leftarrow \neg need2survey, simpleDesignCost$$
$$quote \leftarrow survey, complexDesigCost$$
$$simpleDesignCos$$
$$complexDesignCost$$

Finally, the knowledge of *Surveyor Department* is fairly simple: its domain is its own busyness and since it is lazy it derives that it is busy unless the customer is important

$$busySD \leftarrow not\ important$$

Since such a system must have consistent conclusions we illustrate the results in a weak-weak reasoning. The truth value of the main conclusions are as follows: $important$ is $true_{csd}^{w,w}$, $complexDesignCost$ and $survey$ are $true_{dd}^{w,w}$, $\neg creditWorthy$ is $true_{vc}^{w,w}$; $busySD$ is $false_{sd}^{w,w}$; both $quote$ and $\neg quote$ are $undefined_{csd}^{w,w}$.

Example 7. In this example we illustrate the usage of the proposed framework to reason about evolving knowledge bases. For it, each argumentative agent represents the knowledge (set of rules) added at a point of time. Moreover, each such agents can cooperate with all agents representing past states, and has to argue with all agents representing future states. Consider a concrete example taken from [2] where initially, in Ag_1:

$$sleep \leftarrow not\ tv\_on$$
$$tv\_on \leftarrow$$
$$watch\_tv \leftarrow tv\_on$$

It is easy to see that with this knowledge in Ag_1, there is a justified$_{\mathcal{A}}^{s,w}$ argument for $watch\_tv$ and that the only argument for $sleep$ is overruled$_{\mathcal{A}}^{s,w}$. The knowledge base is then updated, in Ag_2 by adding the rules:

$$\neg tv\_on \leftarrow power\_failure$$
$$power\_failure \leftarrow$$

The reader can check that, for Ag_2 the previous argument for $watch\_tv$ is now overruled, and that the argument for $sleep$ is justified$_{\mathcal{A}}^{s,w}$. Now if another update comes, e.g stating $\neg power\_failure$, in Ag_3 the argument for $sleep$ is again overruled$_{\mathcal{A}}^{s,w}$, and for $watch\_tv$ justified$_{\mathcal{A}}^{s,w}$, as expected. Note how, in this example, the cooperation is used to inherit rules from the past, and the argumentation to make sure that previous rules in conflict with later ones are overruled.

5 Conclusion and Further Work

We propose an argumentation-based negotiation for agent's knowledge bases. We define a declarative semantics for Argumentation-based Negotiation for a multi-agent setting (MAS). Moreover, every agent Ag in a MAS argues and cooperates with a subset of agents in the MAS, i.e. Ag has a set of argumentative agents and a set of cooperative agents. Then, the semantics for Argumentation-based Negotiation is composed by two interlaced processes, viz. argumentation and cooperation. The former imposes that every agent should argue with other agents to evaluate its knowledge. The latter allows an agent to handle its incomplete knowledge with 'help' of other agents. The Argumentation-based Negotiation proposal allows to model a multi-agent setting with different kinds of representation. Furthermore, any agent in a MAS can be queried regarding the truth value of a conclusion. Moreover, a truth value of an agent's belief depends on which agent such a belief is inferred, and also how is the specification of both sets of cooperative and argumentative agents given the overall agents in the MAS. Nevertheless, such answer is always consistent/paraconsistent with the knowledge base of such agents.

Besides the comparisons above, it is worth mentioning [8], which proposes a negotiation framework to be applied in a context of that an agent is a tuple, consisting of its arguments, its domains[9], and lists of its argumentation and cooperation partners. We differ from the authors when they said that an agent should be aware of its domains. A domain is understood by those authors as the set of predicates defining the agent's domain expertise. We assume that our agent has no explicit knowledge about the multi-agent setting domain but we restrict the agents communication by defining the sets of argumentative and cooperative agents. [13] follows the [8]'s specification of an agent. They propose an operational semantics for argumentation by assuming local and global ancestors to detected loop. However, it is not clear how they detected the global ancestor to prove the truth value of an objective literal L.

For the declarative semantics defined here, we have also defined an operational semantics, based on sets of dialogues trees, and an accompanying distributed implementation. The definition of this operational semantics and implementation is however outside the scope of this paper, and is part of a forthcoming one. Also part of ongoing and future work is the comparison of this approach with approaches for dealing with preferences and also with updates in the context of logics programs. It seems clear that the flexibility offered by the sets of cooperative and argumentative agents allows for giving priority to sets of rules over other sets of rules. This is somehow similar to what is done in preferences in the context of logics programs, and a comparison with these frameworks is in order. Also in order is a comparison with logic programming updates. Example 7 suggests how our framework can be used for modelling updates. In this example, the results coincide with those of [2], but a study on how general this equivalence is ongoing.

We also intend to introduce the capability of the agents to revise their knowledge as consequence of internal or external events. In case of no agreement in a negotiation process the agents would be able to discuss (or negotiate again) how and when they got their knowledge, and try to find a way to get an agreement.

[9] A set of predicate names defining the agent's domain expertise.

References

1. J. J. Alferes, C. V. Damásio, and L. M. Pereira. A logic programming system for non-monotonic reasoning. *Journal of Automated Reasoning*, 14(1):93–147, 1995.
2. J. J. Alferes, J. A. Leite, L. M. Pereira, H. Przymusinska, and T. C. Przymusinski. Dynamic updates of non-monotonic knowledge bases. *The Journal of Logic Programming*, 45(1–3):43–70, September/October 2000.
3. A. Bondarenko, P. M. Dung, R. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Journal of Artificial Intelligence*, 93(1–2):63–101, 1997.
4. L. M. Pereira e M. Schroeder C. V. Damásio. Revise: Logic programming and diagnosis. In U. Furbach J. Dix and A. Nerode, editors, *4th International Conference (LPNMR'97)*, volume LNAI 1265 of *Logic Programming and NonMonotonic Reasoning*, pages 353–362. Springer, July 1997.
5. Iara de Almeida Móra and José Júlio Alferes. Argumentative and cooperative multi-agent system for extended logic programs. In F. M. Oliveira, editor, *XIVth Brazilian Symposium on Artificial Intelligence*, volume 1515 of *LNAI*, pages 161–170. Springer, 1998.
6. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Journal of Artificial Intelligence*, 77(2):321–357, 1995.
7. M. Gelfond and V. Lifschitz. Logic programs with classical negation. In Warren and Szeredi, editors, *7th International Conference on LP (ICLP)*, pages 579–597. MIT Press, 1990.
8. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computational*, 8(8):261–292, 1998.
9. L. M. Pereira and J. J. Alferes. Well founded semantics for logic programs with explicit negation. In *European Conference on Artificial Intelligence (ECAI)*, pages 102–106. John Wiley & Sons, 1992.
10. J. L. Pollock. Defeasible reasoning with variable degrees of justification. *Journal of Artificial Intelligence*, 133:233–282, 2002.
11. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.
12. H. Prakken and G. A. W. Vreeswijk. *Handbook of Philosophical Logic*, volume 4, chapter Logics for Defeasible Argumentation, pages 218–319. Kluwer Academic, 2 edition, 2002.
13. M. Schroeder and R. Schweimeier. Arguments and misunderstandings: Fuzzy unification for negotiating agents. *Journal of Computational Logic in Multi-Agent Systems*, 93:1–18, August 2002.
14. Michael Schroeder, Iara de Almeida Móra, and J. J. José Júlio Alferes. Vivid agents arguing about distributed extended logic programs. In Ernesto Costa and Amilcar Cardoso, editors, *Progress in Artificial Intelligence, 8th Portuguese Conference on Artificial Intelligence (EPIA)*, volume 1323 of *LNAI*, pages 217–228. Springer, 1997.
15. R. Schweimeier and M. Schroeder. Notions of attack and justified arguments for extended logic programs. In F. van Harmelen, editor, *15th European Conference on Artificial Intelligence*. IOS Press, 2002.
16. G. A. W. Vreeswijk. Abstract argumentation systems. *Journal of Artificial Intelligence*, 90(1–2):225–279, 1997.