Argumentative and Cooperative Multi-agent System for Extended Logic Programming*

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

[1] CENTRIA, Univ. Nova de Lisboa
2825 Monte da Caparica, Portugal
{idm,jja}@di.fct.unl.pt
[2] Dept. Matemática, Univ. de Évora
R. Romão Ramalho 59
7000 Évora, Portugal
jja@dmat.uevora.pt

Abstract. The ability to view extended logic programs as argumentation systems opens the way for the use of this language in formalizing communication among reasoning computing agents in a distributed framework. In this paper we define an argumentative and cooperative multi-agent framework, introducing credulous and sceptical conclusions. We also present an algorithm for inference and show how the agents can have more credulous or sceptical conclusions.

1 Introduction

The notion of autonomous agent is used to denote the fact that an agent has the ability to decide for itself which goals it should adopt and how these goals should be achieve. In most agents applications the autonomous components need to interact with one another given the inherent interdependencies which exist between then. The predominant mechanism for managing these interdependencies at run-time is negotiation. Negotiation is a process that takes place between two or more agents which communicate to try and to come to a mutually acceptable agreement on some matter to achieve goals which they cannot, or prefer not to, achieve on their own. Their goals may conflict – in which case the agents have to bargain about which agent achieve which goal – or the agents may depend upon each other to achieve the goals.

It is our conviction that the use of a particular argumentation system provides the negotiation mechanism. Argument-based systems analyze defeasible, or non-monotonic reasoning in terms of the interactions between arguments to get an agreement of a goal. Non-monotonicity arises from the fact that arguments can be defeated by stronger (counter)arguments. The intuitive notions of argument, counterargument, attack and defeat have natural counterparts in the real world, which makes argument-based systems suitable for multi-agents

* Thanks to Luis Moniz Pereira and Michael Schroeder for their valuable comments. The work is financially supported by the Portuguese JNICT and Brazilian CAPES.

applications. The ability to view extended logic programs as argumentation systems opens the way for the use of this language in formalizing communication among reasoning computing agents in a distributed framework. Argumentation semantics in extended logic programming has been defined in [3, 4, 8, 10] for an agent, determining the agent's beliefs by an internal argumentation process. Intuitively, argumentation semantics treats the evaluation of a logic program as an argumentation process, where a goal G holds if all arguments supporting G cannot be attacked. Thus, logic programming can be seen as a discourse involving attacking and counter-attacking arguments.

Dung [3, 4] defines whether an argument is acceptable or not with respect to a set of arguments. Prakken and Sartor [8] follow Dung's proposal and introduce a third kind of argument status, defining when an argument is justified, overruled or defeasible wrt. a set of arguments. Both argumentation frameworks have different methods to obtain credulous or sceptical results. Móra, Schroeder and Alferes's [10] modify Prakken and Sartor's proposal and define a more credulous argumentation framework. While Dung uses argumentation to define a declarative semantics for extended logic programs (ELP), Prakken and Sartor's work is driven by applications in legal reasoning. Both Dung's and Prakken and Sartor's work refers to argumentation in a single agent. In [10, 6] we generalize their work to a multi-agent setting. In this paper we present a negotiation framework more flexible than the previous ones that allows the agents to obtain conclusions in a more credulous or sceptical way, depending on the application's goals.

1.1 Extended Logic Programming

In the sequel we use the language of Extended Logic Programming, a generalization of Normal Logic Programming with explicit negation [5]:

Definition 1 (Extended Logic Program). *An* extended logic program *(ELP) is a (possibly infinite) set of ground rules*[1] *of the form* $L_0 \leftarrow L_1, \ldots, L_l, not\ L_{l+1}, \ldots, not\ L_m$ $(0 \leq l \leq m)$ *where each* L_i *is an objective literal* $(0 \leq i \leq m)$. *If* $n = 0$ *then the rule is called a* fact *and the arrow symbol is omitted. An objective literal is either an atom* A *or its explicit negation* $\neg A$. *Literals of the form* $not\ L$ *are called default literals. As usual* L_0 *is called the* head, *and* $L_1, \ldots, not\ L_n$ *the* body *of the rule.*

For this language, various declarative semantics have been defined, e.g. the answer-sets semantics [5] (which is a generalization of the stable models semantics of normal logic programs), the well-founded semantics with explicit negation (WFSX) [7], and the well-founded semantics with "classical" negation [9]. Both of the last two are generalization of the well-founded semantics of normal programs. However, the former, unlike the latter, considers the so-called coherence requirement relating the two form of negation: *"if L is explicitly false then L must*

[1] For simplicity, we use non-grounded rules in some examples. These rules simply stand for its ground version, i.e. for the ground rules obtained by substituting in all possible ways each of the variables by elements of the Herbrand Universe.

be assumed false by default". A paraconsistent extensions of WFSX ($WFSX_p$) has been defined in [1]. In it, unlike in the others mentioned above, contradictory information is accepted and deal with by the semantics. For further details on the subject of extended logic programs, their semantics, applications, and implementation see [2].

1.2 Further Motivation

A negotiation process is started by an agent that should prove some goal to answer a request from another agent, or to satisfy some internal decision. Intuitively, we see the negotiation process as a dialogue. A negotiative dialogue is an exchange of arguments between two (or more) agents, where each alternately presents arguments attacking the previous proposed arguments. The agent who fails to present (counter)arguments looses the negotiation. A process of negotiation is started by an agent who puts forward an initial proposal, i.e. an argument for some goal that should be proven or achieved. Then the receipts agents evaluate the proposal. Arguments can be unacceptable either by having conflicting goals or by having incomplete information. In the former case, the recipient must construct a counterargument to attack the initial proposal (argumentation). In the latter, the try to construct an argument in favor of the initial proposal (cooperation). When the proposed argument is acceptable, the recipient just agree with the initial proposal. The negotiation process continues until an argument or a counterargument is acceptable to all the parties involved, or until the negotiation breaks down without an agreement.

An interesting situation that can happen in a multi-agent negotiation is when an argument proposed to an agent causes contradiction in its knowledge (i.e. the agent concludes both L and $\neg L$). Consider now a variation of the well known Tweety example where an agent Ag_1 knows that "Tweety is a penguin", "Birds normally fly" and "Penguins do not fly", and another agent, Ag_2, knows that "Tweety is a bird". The agents' knowledge can be represented by logic programs and the sequence of rules can be seen as arguments by the corresponding agents:

$$Ag_1 = \{p(t); f(X) \leftarrow b(X), not\ ab(X); \neg f(X) \leftarrow p(X)\} \quad Ag_2 = \{b(t)\}$$
$$Ag_1\colon [\neg f(t) \leftarrow p(t); p(t)]\ (A_1) \quad Ag_1\colon [p(t)]\ (A_2)$$
$$Ag_1\colon [f(t) \leftarrow b(t); b(t)] \quad (A_3) \quad Ag_2\colon [b(t)]\ (A_4)$$

Then a negotiation for "Tweety is a bird" starts by Ag_2 proposing to Ag_1 the argument A_4. Ag_1 has no counterargument for A_4 but Ag_1 cannot agree with $b(t)$ because its addition to Ag_1's knowledge would cause a contradiction (Ag_1 would conclude both $f(t)$ and $\neg f(t)$). In those an argument is not accepted if it has some hypotheses that can be defeated by an agent. This is not the case here, where the argument that is not acceptable does not even have hypotheses. We can be more flexible in this assumption by defining sceptical or credulous agents. Sceptical agents neither agree with a goal L nor a goal $\neg L$, even if one of these is a fact. Credulous agents may agree with L even if other agents (or itself) also agree with $\neg L$. Then, if Ag_1 is a credulous agent, it will accept the fact that "Tweety is a bird". If the agent is sceptical, it will not agree with that fact. Note that this situation happens because there is no evidence "Tweety is an abnormal bird" to defeat "Tweety flies since it is a bird and *there is no evidence* that it is an abnormal bird". In this paper we do not consider a "belief revision" approach to solve situations when there is no agreement between agents. In such approach, the contradiction between fly and $\neg fly$ could be solved if Ag_1 assumes the objective literal ab.

Next we define an argumentative and cooperative multi-agent framework by combining and extending the proposal of [6]. Finally, we present an algorithm for inference and show how the agents can have more credulous or sceptical conclusions.

2 Multi-agent Negotiation

A multi-agent system is a set of several extended logic programs, each corresponding to an individual agent with an individual view of the world. Agents negotiate by exchanging parts of several independent or overlapping programs and should obtain consensus concerning inference of literals:

Definition 2 (Multi-agent System). *Let Ag_i be extended logic programs, where $1 \leq i \leq n$. The set $\mathcal{A} = \{Ag_1, \ldots, Ag_n\}$ is called a* **M**ulti-**A**gent **S**ystem.

An agent Ag negotiates, i.e. argues and cooperates, by using arguments. An argument is a sequence of ground rules chained together, based on facts and where all negative literals are viewed as hypotheses and assumed to be true. Our negotiation framework has two kinds of arguments, strong arguments and weak arguments. A weak argument is more liable to be attacked than a strong one. Intuitively, a strong argument A_L (for some literal L) can only be attacked by "proving" the complement of a (negative) hypothesis assumed in A_L. A weak argument A'_L (for some literal L) can also be attacked by "proving" the explicit complement[2] of a conclusion of A'_L. For example, if "Tweety flies because it is a bird, non-abnormal birds fly, and I assume that Tweety is not abnormal" is a strong argument, it can be only attacked by "proving" that Tweety is abnormal. If it were a weak argument, it could also be attacked by "proving" that Tweety does not fly.

Definition 3 (Complete Strong and Weak Arguments). *Let Ag be an extended logic program. A* strong *(resp.* weak*)* argument *for L is a finite sequence A^s_L (resp. A^w_L) $= [r_n; \ldots; r_m]$ of rules r_i of the form $L_i \leftarrow Body_i$ (resp. $L_i \leftarrow Body_i, not\ \neg L_i$) such that $L_i \leftarrow Body_i \in Ag$ and (1) for every $n \leq i \leq m$, and every objective literal L_j in the body of r_i there is a $k < i$ such that L_j is the head of r_k; (2) L is the head of some rule of A^s_L (resp. A^w_L); and (3) no two distinct rules in the sequence have the same head. We say A_L is an argument for L if it is either a strong or a weak argument for L. An argument A_L is a* sub-argument *of the argument $A_{L'}$ iff A_L is a prefix of $A_{L'}$.*

[2] By the explicit complement of an objective literal L we mean $\neg L$, if L is an atom, or A if $L = \neg A$.

However, if an agent Ag has no complete knowledge of a literal L then Ag has just a partial argument for L:

Definition 4 (Strong and Weak Partial Argument). *Let Ag be an extended logic program. A* strong *(resp.* weak*) partial argument of L is a finite sequence PA_L^s (resp. PA_L^w) $= [r_n; \ldots; r_m]$ of rules r_i of the form $L_i \leftarrow Body_i$ (resp. $L_i \leftarrow Body_i, not\ \neg L_i$) such that $L_i \leftarrow Body_i \in Ag$ and (1) for every $n < i \leq m$, and every objective literal L_j in the body of r_i there is a $k < i$ such that L_j is the head of r_k; (2) L is the head of some rule of PA_L^s (resp. PA_L^w); and (3) no two distinct rules in the sequence have the same head. We say PA_L is a partial argument for L if it is either strong or weak partial argument for L.*

Example 1. Based on the example presented on section 1.2, the set $\mathcal{A} = \{Ag_1, Ag_2\}$ is a multi-agent system such that $Ag_1 = \{p(t); f(X) \leftarrow b(X), not\ ab(X); \neg f(X) \leftarrow p(X)\}$ and $Ag_2 = \{b(t)\}$. The set of arguments[3] of Ag_1 is $Arg_1 = \{[p], [p \leftarrow not\ \neg p], [p; \neg f \leftarrow p], [p \leftarrow not\ \neg p; \neg f \leftarrow p, not\ f]\}$, i.e. $\{A_p^s, A_p^w, A_{\neg f}^s, A_{\neg f}^w\}$, and of Ag_2 is $Arg_2 = \{[b]; [b \leftarrow not\ \neg b]\}$, i.e. $\{A_b^s, A_b^w\}$, and the partial arguments for f are $PA_f^s = \{[f \leftarrow b, not\ ab]\}$ and $PA_f^w = \{[f \leftarrow b, not\ ab, not\ \neg f]\}$.

An argument can be attacked by contradicting (at least) one hypothesis in it. Since arguments assume some negative literals (hypotheses), another complete argument for its complement attacks the former. In this respect, our notion of attack differs from the one of Prakken and Sartor [8] and of Dung [3]. In theirs, an argument attacks another argument via rebut or undercut [8] (resp. RAA- or ground-attack in [3]). The difference depends on whether the attacking argument contradicts a conclusion[4], in the former, or an assumption of another argument[5], in the latter. Since our argumentation framework has two kinds of arguments, the strong and the weak, the first kind of attack is not needed. Simply note that rebut are undercut attacks to weak arguments. Moreover, partial arguments of an agent may be completed with the help of arguments of other agents (i.e. agents may *cooperate* to complete arguments). Such partial arguments can also be used to attack other arguments.

Definition 5 (Direct Attack). *Let A_L and $A_{L'}$ be strong or weak arguments, A_L* attacks *$\{A_{L'}\}$ iff (1) A_L is an argument for L and $A_{L'}$ is an argument with assumption not L, i.e. there is a $r : L_0 \leftarrow L_1, \ldots, L_l, not\ L_{l+1}, \ldots, not\ L_m \in A_{L'}$ and a j such that $L = L_j$ and $l+1 \leq j \leq m$; or (2) A_L is a partial argument for L, $A_{L'}$ has the assumption not L, and there exists a sequence of arguments $A_{L_1}, \ldots, A_{L_n}$ such that $A_L + A_{L_1} + \ldots + A_{L_n}$[6] is a complete argument for L.*

[3] Assuming that the Herbrand universe only has one individual, we suppress t from the literals, i.e. f stands for $f(t)$.

[4] An argument A_L rebuts/RAA-attacks $A_{\neg L}$, and vice-versa.

[5] An argument A_L undercuts/g-attacks $A_{L'}$ if it contains the default literal *not* L.

[6] By $A + B$ we mean the concatenation of arguments A and B.

Note that the second point of the definition may be viewed as cooperation among some agents with the purpose of another agent. Not all arguments make sense. For example, it does not make sense to have an argument which assumes some hypothesis and, at the same time, concludes its negation. The notions of coherent argument and conflict-free set of arguments formalize what we mean by arguments and sets of arguments that "make sense":

Definition 6 (Coherent, Conflict-free). *An argument A_L is* coherent *if it does not contain sub-arguments attacking each other, i.e. there are no two sub-arguments in A_L, $A_{L'}$ and $A_{L''}$, such that $A_{L'}$ is a complete argument for a literal L' and $A_{L''}$ is an argument with assumption not L'. A set of arguments $Args$ is called* conflict-free *if no two arguments in $Args$ attack each other.*

This attack definition does not completely foresees the cases where an argument causes conflict among arguments, as shown by the example below. This is the motivation for the definition of indirect attacks.

Example 2. Continuing example 1, Arg_1 is a conflict-free set of (partial and complete) arguments, $\{A_p^s, A_p^w, A_{\neg f}^s, A_{\neg f}^w, PA_f^s, PA_f^w\}$. Suppose that agent Ag_2 proposes the argument A_b^w to Ag_1. Intuitively we saw that A_b^w *indirectly attacks* the subset of arguments $S = \{PA_f^w, A_{\neg f}^w\}$ of Ag_1. Note that PA_f^w is a partial argument because it has no sub-argument for b, but $PA_f^w + A_b^w$ is now an argument for f, i.e. $A_f^w = [b \leftarrow not\ \neg b; f \leftarrow b, not\ ab, not\ \neg f]$.

Definition 7 (Indirect Attack). *Let $Args$ and $Args'$ be conflict-free sets of arguments, and $A_{L'}$ be an argument in $Args'$. $A_{L'}$* indirectly attacks *a subset S of $Args$ iff $S \cup \{A_{L'}\}$ is not conflict-free. An argument A_L attacks a set of arguments S iff A_L directly or indirectly attacks S.*

To defeat a set of arguments S, an agent needs an argument A attacking at least one of the arguments in S that, in turn, is not itself attacked by arguments in S. This, unless the set S contains some incoherent argument, in which case nothing else is needed to defeat S.

Definition 8 (Defeat). *Let $Args$ be a set of arguments, S be a subset of $Args$, and A_L and $A_{L'}$ be arguments. $A_{L'}$* defeats *S iff (1) $A_{L'}$ directly attacks S and there is no $A_L \in S$ that attacks $\{A_{L'}\}$; or (2) $A_{L'}$ is empty and there is some incoherent argument $A_L \in S$.*

An argument is acceptable if it can be defended against all attacks, i.e. all arguments that attack it can be defeated. Since we have defined strong and weak arguments, the definition of acceptable (and corresponding characteristic function) [4] can be generalized by parameterizing the kinds of arguments:

Definition 9 ($Acceptable_o^p$). *An argument A_L is $acceptable_o^p$ wrt. a set of arguments $Args$ iff each o argument $A_{L'}$ attacking A_L is defeated by a p argument in $Args$.*

Definition 10 (Characteristic Function). *Let Ag be an extended logic program in $\mathcal{A}$, $Args$ be the set of arguments of $\mathcal{A}$ and $S \subseteq Args$. The* characteristic function *of Ag and S is $F_{Ag}(S) = \{A_L \in S \mid A_L$ is $acceptable_o^p$ wrt. $S\}$.*

2.1 The status of an argument and agent's conclusions

The status of an argument is determined based on all ways in which arguments interact. It takes as input the set of all possible arguments and their mutual defeat relations, and produces as output a split of arguments into three classes: arguments with which a dispute can be 'won', 'lost', and arguments which leave the dispute undecided. The "weakest link" principle defines that an argument cannot be justified unless all of its sub-arguments are justified. We define the losing, or "overruled" arguments as those that are attacked by a justified argument and, finally, the undeciding, or "defeasible" arguments as those that are neither justified nor overruled.

Definition 11 (Justified, Overruled and Defeasible). *Let Ag be an extended logic program in $\mathcal{A}$, $Args$ be the set of arguments of $\mathcal{A}$, A_L be an argument of Ag, and F_{Ag} be the characteristic function of Ag and $Args$ then (1) A_L is* justified *iff it is in the least fixpoint of F_{Ag} (called $JustArgs$); (2) A_L is* overruled *iff it is attacked by a justified argument; and (3) A_L is* defeasible *iff it is neither justified nor overruled.*

Based on the status of the arguments we are able to determine the possible values for these conclusions. Note that with weak and strong arguments, a given literal might have more that one conclusion. If both (weak and strong) arguments for L have the same status, i.e. both are justified, overruled or defeasible, then both conclusions for L also coincide. If the strong and the weak argument do not coincide and the values are inconsistent, i.e. the strong argument is justified and the weak is overruled, then a credulous agent would conclude that the literal is justified, and a sceptical one that is defeasible. Otherwise, a credulous agent assumes the value of the strong argument and a sceptical one assumes the value of the weak argument.

Definition 12 (Sceptical and Credulous Conclusion). *Let A_L be an argument for L, C_L^s be a sceptical conclusion for L and C_L^c a credulous one, and C_L be either C_L^s or C_L^c then (1) if A_L is justified (resp. overruled, defeasible) then C_L is justified (resp. overruled, defeasible); (2) if A_L^s is justified and A_L^w is overruled then C_L^c is justified and C_L^s is defeasible; (3) if A_L^s is justified (resp. defeasible) and A_L^w is defeasible (resp. overruled) then C_L^c is justified (resp. defeasible) and C_L^s is defeasible (resp. overruled).*

Example 3. The rule $r_1 : r(X) \leftarrow not\ d(X)$ states that "Someone is republican if it is not democrat" and the rule $r_2 : d(X) \leftarrow not\ r(X)$ states that "Someone is democrat if it is not republican". The fact $f_1 : \neg d(john)$ states that "John is not a democrat". The set of arguments, with obvious abbreviations, is $Args = \{[\neg d], [\neg d \leftarrow not\ d], [d \leftarrow not\ r], [d \leftarrow not\ r, not\ \neg d], [r \leftarrow not\ d], [r \leftarrow not\ d, not\ \neg r]\}$, i.e. $\{A^s_{\neg d}, A^w_{\neg d}, A^s_d, A^w_d, A^s_r, A^w_r\}$. The arguments $A^s_{\neg d}$ and A^s_r are *justified*, A^s_d is *overruled*, and the *defeasible* ones are $A^w_{\neg d}$, A^w_d and A^w_r. Then the credulous conclusion (wrt. $Args$) for $\neg d$ and r is both are justified, and d is overruled. The sceptical conclusion (wrt. $Args$) assumes all literals as defeasible.

Continuing the example 1, the credulous conclusion (wrt. $Args$) for b, p, f, $\neg f$ is that all are justified. However, the sceptical conclusion (wrt. $Args$) is that b and p are justified, and f and $\neg f$ are defeasible.

3 Proof proposal for an argument

Based on Praken and Sartor's dialogue proposal [8], we propose a (credulous and sceptical) proof for a justified argument as a dialogue tree where each branch of the tree is a dialogue. As seen in section 2, to attack others' arguments, an agent may need to cooperate in order to complete its partial arguments. To account for this, we start by defining a cooperative dialogue. Intuitively, in cooperative dialogues other agents provide arguments for the still incomplete argument. The process stop either if no agent can provide the required arguments or if all arguments are completed.

Definition 13 (Cooperative Dialogue). *Let $\mathcal{A}$ be a MAS, Ag be an agent in $\mathcal{A}$, $A_L = [r_n; \ldots; r_m]$ be a weak or strong partial argument of Ag, and Inc be the set of objective literals L_j in the body of r_n. A* cooperative dialogue *for A_L in Ag is a finite nonempty sequence of moves, $move^c_i = (Ag_k, A_{L_i})$ $(i > 0)$ such that (1) $move^c_1 = (Ag, A_L)$. (2) If A_L is a weak (resp. strong) partial argument then A_{L_i} is a weak (resp. strong) argument. (3) If $i \neq j$ then $S_i \neq S_j$. (4) Let $move^c_i = (Ag_i, [r_n; \ldots; r_m])$ then (a) if there exists an $Ag_k \in \mathcal{A}$ with an argument for some $L \in Inc$, $move^c_{i+1} = (Ag_k, [r_{n_k}; \ldots; r_{m_k}; r_n; \ldots; r_m])$; otherwise, (b) $move^c_i$ is the last move in the dialogue. A cooperative dialogue of an agent Ag for a partial argument A_L is successful if its last move (Ag_f, A_{L_f}) is a (complete) argument. Then we call A_{L_f} its* result.

An (argumentative) dialogue is made between a proponent P and opponents O, and the root of the dialogue tree is an argument for some literal L. An (argumentative) move in a dialogue consists of an argument attacking the last move of the other player. The required force of a move depends on who states it. Since the proponent wants a conclusion to be justified, a proponent's argument has to be defeating while, since the opponents only want to prevent the conclusion from being justified, an opponent move may be just attacking. Based on our definition of credulous and sceptical conclusions, we define a (flexible) dialogue that allow us to obtain conclusions in a more credulous or sceptical way. It depends on how the players, i.e. a proponent P and opponents O, move their arguments. A $dialogue^p_o$ is a dialogue between a proponent that supports p conclusions and an opponent that supports o conclusions; p and o represent a strong (s) or a weak (w) conclusion:

Definition 14 ($dialogue^p_o$). *Let $\mathcal{A}$ be a MAS, Ag be an agent in $\mathcal{A}$, S be a subset of arguments of Ag, and p (resp. o) be a weak or a strong conclusion from the proponent (resp. opponent). A $dialogue^p_o$ for an argument A_L of an agent Ag is a finite nonempty sequence of (argumentative) moves $move^a_i = (Role_i, Ag_i, S_i)$ $(i > 0)$, where $Role_i \in \{P, O\}$ and $Ag_i \in A$, such that (1) $move^a_1 = (P, Ag, A_L)$.*

(2) $Role_i = P$ *iff* i *is* odd*; and* $Role_i = O$ *iff* i *is* even*. (3) If* $Player_i = P$ *then every* $A_{L_i} \in S_i$ *supports* p *conclusions; otherwise, they support* o *conclusions. (4) If* $Role_i = Role_j = P$ *and* $i \neq j$*, then* $S_i \neq S_j$*. (5) Every* $A_{L_i} \in S_i$ *are either arguments of* Ag_i *or the* result *of a successful cooperation dialogue for partial arguments of* Ag_i*. (6) If* $Role_i = P$ $(i > 1)$*, then every* $A_{L_i} \in S_i$ *are minimal (wrt. set inclusion) arguments defeating some* $A_{L'} \in S_{i-1}$*. (7) If* $Role_i = O$*, then* S_i *attacks some* $A_{L'} \in S_{i-1}$*.*

A dialogue tree considers all ways in which an agent can attack an argument:

Definition 15 (Dialogue Tree). *Let* $\mathcal{A}$ *be a MAS,* Ag *be an agent in* $\mathcal{A}$*,* S *be a set of arguments, and* A_L *be an argument of* Ag*. A* dialogue tree *for* A_L *is a finite tree of moves* $move_i^a = (Role, Ag_k, S_i)$ *such that each branch is a* $dialogue_o^p$ *for* A_L*, and if* $Role = P$ *then the children of* $move_i^a$ *are all defeaters of an argument* $A_{L'} \in S_i$*. An agent wins a dialogue tree iff it wins all branches of the tree, and an agent wins a* $dialogue_o^p$ *if another agent cannot counter-argue.*

Proposition 1. *Let* Ag *and* Ag_k *be agents, and* A_L *be an argument of* Ag*. An argument* A_L *is* justified *iff* Ag *wins the dialogue tree starting with* A_L*; otherwise,* A_L *is overruled or defeasible. An argument* A_L *is overruled when an opponent* Ag_k *moves at least one justified argument; otherwise,* A_L *is defeasible.*

In the extended version of this article [7] we continue by showing how the players should negotiate, i.e. how they should move their arguments in order to obtain more credulous or sceptical conclusions, and what is correspondence with other semantics. We define which kind of (strong or weak) argument the proponent should move to justify a credulous or a sceptical conclusion, and what kind of (strong or weak) opponent's argument the proponent should defeat to win a negotiation, and compare the results with others. Lack of space prevents us from presenting here the full results, but a brief overview of those is presented below.

To justify a sceptical conclusion, the proponent has to justify both arguments for L. However, it is sufficient to justify only the weak argument. To win the argumentation, the proponent has to defeat all opponent's arguments, i.e. to defeat the strong and the weak ones. Nevertheless the proponent should defeat just the opponent's weak argument. Then both players have to move weak arguments to get a sceptical conclusion. The $dialogue_o^p$ has the same results as Prakken and Sartor's dialogue proposal [8].

The dialogue is an paraconsistent (and credulous) semantics if a player concludes L and if it is also capable of concluding $\neg L$. To obtain a credulous conclusion, the proponent has to justify a single argument in order to prove the conclusion. Thus, it is sufficient for it to justify the strong argument. Similarly to the sceptical proof, the proponent only has to defeat the weak argument. The results of $dialogue_w^s$ are the same as those obtained by the $WFSX_p$ semantics [1] on the union of all agents' programs. Finally, if both players, the proponent and the opponent, support strong conclusions the semantics coincides with the WFS with "classical negation" defined in [9].

[7] In "http://www-ssdi.di.fct.unl.pt/ idm/fullpaper/mora-ExtSBIA98.ps.gz".

4 Conclusion and Further work

In this paper we propose a negotiation framework for a multi-agent system, based on argumentation and cooperation. We define strong and weak arguments, simplify the notion of ground- and RAA-attack [3] (resp. undercut and rebut [8]), propose a new concept of attack (indirect attacks), and introduce the credulous and the sceptical conclusions for an agent's literal. Finally, we present inference algorithms to obtain credulous and sceptical conclusions and we show the relation of these algorithms are similar to Prakken and Sartor's argumentation [8], and also to the known semantics, $WFSX$ [7] and $WFSX_p$ [1].

We intend now to define a multi-agent system where the agents have several degrees of credulity, from agents that always accept new information to agents that accept the new one only if it is not incompatible with agent knowledge. 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.

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 and L. M. Pereira. *Reasoning with Logic Programming.* (LNAI 1111), Springer, 1996.
3. P. M. Dung. An argumentation semantics for logic programming with explicit negation. In *10th ICLP*, pages 616–630. MIT Press, 1993.
4. P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
5. M. Gelfond and V. Lifschitz. Logic programs with classical negation. In Warren and Szeredi, editors, *7th ICLP*, pages 579–597. MIT Press, 1990.
6. I. A. Móra and J. J. Alferes. Argumentation and cooperation for distributed extended logic programs. In J.Dix and J.Lobo, editors, *7th International Workshop on Nonmonotonic Reasoning (NMRW)*, 1998.
7. L. M. Pereira and J. J. Alferes. Well founded semantics for logic programs with explicit negation. In *ECAI*, pages 102–106. John Wiley & Sons, 1992.
8. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 1997.
9. T. Przymusinski. Extended stable semantics for normal and disjunctive programs. In Warren and Szeredi, editors, *7th ICLP*, pages 459–477. MIT Press, 1990.
10. M. Schroeder, I. A. Móra, and J. J. 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 (EPIA97)*, pages 217–228. (LNAI 1323) Springer. Coimbra, Portugal, 1997.