Dialoguing DeLP-based agents

Diego C. Martínez [*] Alejandro García
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Av. Alem 1253 - (8000) Bahía Blanca - Buenos Aires - República Argentina
Tel/Fax: (+54)(291)4595135/5136 - E-mail: **{dcm,ajg}@cs.uns.edu.ar**

Abstract

A multi-agent system is made up of multiple interacting autonomous agents. It can be viewed as a society in which each agent performs its activity cooperating to achieve common goals, or competing for them. They establish dialogues via some kind of agent-communication language, under some communication protocol. We think argumentation is suitable to model several kind of dialogues in multi-agents systems. In this paper we define dialogues and persuasion dialogues between two agents using Defeasible Logic Programs as a knowledge base, together with an algorithm defining how this dialogue may be engaged. We also show an indication of how an agent could use opponent's information for its own benefit.

1 Introduction

A multi-agent system is made up of multiple interacting autonomous agents. It can be viewed as a society in which each agent performs its activity cooperating to achieve common goals, or competing for them. These agents need to interact with one another because of the inherent interdependencies which exist between them [3]. They should have the ability to establish dialogues via some kind of agent-communication language, under some communication protocol. As stated in [1], we think argumentation is suitable to model several kind of dialogues in multi-agents systems. It's being used as a form of negotiation between agents, where they communicate to one another to try to come to a mutually acceptable agreement on some matter [7].

Argumentation can be considered as a process in which the parties (in this case, agents) exchange arguments for or against some proposition, usually in order to persuade each other. An argument is a subjective explanation of some statement being

[*] Becario de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

alleged in that process. In the next section we define *Defeasible Logic Programs*, a well-known formalism of non-monotonic reasoning. These programs are used to represent the knowledge base of an agent.

2 Defeasible Logic Programs

Defeasible Logic Programming is a logic programming paradigm based on a defeasible argumentation formalism that allows the representation of strict and defeasible knowledge

In DeLP, a literal h is *warranted* if there exists a non-defeated *argument* $\mathcal{A}$ supporting h. An argument $\langle \mathcal{A}, h \rangle$ for a literal h is a minimal and consistent set of defeasible rules that allows to infer h. In order to establish whether $\langle \mathcal{A}, h \rangle$ is a non-defeated argument, *argument rebuttals* or *counter-arguments* that could be *defeaters* for $\langle \mathcal{A}, h \rangle$ are considered, *i.e.*, counter-arguments that by some criterion, are preferred to $\langle \mathcal{A}, h \rangle$. Since counter-arguments are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments *called argumentation line* may appear, where each argument defeats its predecessor in the line (see the following example). Usually, each argument has more than one defeater and more than one argumentation line exists. Therefore, a tree of arguments called *dialectical tree* is constructed, where the root is $\langle \mathcal{A}, h \rangle$ and each path from the root to a leaf is an argumentation line. A *dialectical analysis* of this tree is used for deciding whether h is warranted.

The interested reader is referred to [8, 9, 10] for details about DeLP.
Defeasible logic programs are defined in terms of two types of rules:

- *stric rules* $l \ \leftarrow \ q_1, \ldots, q_n.$, used to represent not defeasible information and
- *defeasible rules* $l \prec q_1, \ldots, q_n$, used to represent defeasible knowledge,

where l is a literal (i. e., a predicate "p" or a negated predicate "$\sim p$"), and each q_i $(n \geq 0)$ is a literal, or a literal preceded by the symbol not. A *defeasible logic program* (*de.l.p.*) is a finite set of defeasible and strict rules. Given a *de.l.p.* $\mathcal{P}$, a *defeasible derivation* for a literal q is a finite set of defeasible and strict rules obtained by backward chaining from q as in a PROLOG program, using both strict and defeasible rules. The subset of all extended program clauses must be *consistent*, i.e., there is no defeasible derivation of complementary literals, although the subset of defeasible clauses may be inconsistent. It is only in this form that a *de.l.p.* may contain potentially inconsistent information. In this framework, an argument is considered a defeasible reason for supporting conclusions.

Definition 2.1 (Argument Structure). *[8] Let h be a literal, and $\mathcal{P}$=(Π, Δ) a de.l.p.. We say that $\langle \mathcal{A}, h \rangle$ is an argument structure for h, if $\mathcal{A}$ is a set of defeasible rules of Δ, such that:*

1. *there exists a defeasible derivation for h from $\Pi \cup \mathcal{A}$,*
2. *the set $\Pi \cup \mathcal{A}$ is non-contradictory, and*
3. *$\mathcal{A}$ is minimal: there is no proper subset $\mathcal{A}'$ of $\mathcal{A}$ such that $\mathcal{A}'$ satisfies conditions (1) and (2).*

Since the subset of defeasible clauses may be inconsistent, the set of all arguments which can be made from a *de.l.p.* will conflict. This fact lead us to the following definitions of opposing arguments:

Definition 2.2 (Counter-argument). *[8] We say that $\langle \mathcal{A}_1, h_1 \rangle$* counter-argues, rebutts, *or* attacks *$\langle \mathcal{A}_2, h_2 \rangle$ at literal h, if and only if there exists a sub-argument $\langle \mathcal{A}, h \rangle$ of $\langle \mathcal{A}_2, h_2 \rangle$ such that h and h_1* disagree*.*

Definition 2.3 (Defeating argument). *[8]*
An argument $\langle \mathcal{A}_1, h_1 \rangle$ defeats *an argument $\langle \mathcal{A}_2, h_2 \rangle$ at literal h, if and only if there exists a sub-argument $\langle \mathcal{A}, h \rangle$ of $\langle \mathcal{A}_2, h_2 \rangle$ such that $\langle \mathcal{A}_1, h_1 \rangle$ counterargues $\langle \mathcal{A}_2, h_2 \rangle$ at h, and either:*
(1) $\langle \mathcal{A}_1, h_1 \rangle$ is strictly better [1] than $\langle \mathcal{A}, h \rangle$
(2) $\langle \mathcal{A}_1, h_1 \rangle$ is incomparable to $\langle \mathcal{A}, h \rangle$

As stated before, any conclusion h will be considered warranted only when the argument that supports it becomes a *justification*. To check if an argument A is a justification for a literal h we need to start an analysis between A and its defeaters. Since defeaters are arguments, there may exist defeaters for defeaters, and so on. This recursive justification process is called a *dialectical analysis* and can be represented as trees in the following manner. The root of the tree will correspond to the initial argument, and every inner node will represent a defeater (proper or blocking) of its father. Leaves in this tree will correspond to non-defeated arguments [9].

Definition 2.4 (Dialectical tree [8]).
Let $\mathcal{A}$ be an argument for h. A dialectical tree *for $\langle \mathcal{A}, h \rangle$, denoted $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$, is recursively defined as follows:*
(1) A single node labeled with an argument $\langle \mathcal{A}, h \rangle$ with no defeaters (proper or blocking) is by itself a dialectical tree for $\langle \mathcal{A}, h \rangle$. This node is also the root of the tree.
(2) Suppose that $\langle \mathcal{A}, h \rangle$ is an argument with defeaters (proper or blocking) $\langle \mathcal{A}_1, h_1 \rangle$, $\langle \mathcal{A}_2, h_2 \rangle$, ..., $\langle \mathcal{A}_n, h_n \rangle$. We construct the dialectical tree for $\langle \mathcal{A}, h \rangle$, $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$, by labeling the root node of with $\langle \mathcal{A}, h \rangle$ and by making this node the parent node of the roots of the dialectic trees for $\langle \mathcal{A}_1, h_1 \rangle$, $\langle \mathcal{A}_2, h_2 \rangle$, ..., $\langle \mathcal{A}_n, h_n \rangle$, i. e., $\mathcal{T}_{\langle \mathcal{A}_1,\, h_1 \rangle}$, $\mathcal{T}_{\langle \mathcal{A}_2,\, h_2 \rangle}$, ..., $\mathcal{T}_{\langle \mathcal{A}_n,\, h_n \rangle}$.

Nodes in the dialectical tree can be recursively marked as *defeated* or *undefeated* nodes (D-nodes and U-nodes respectively). Let $\mathcal{A}$ be an argument for a literal h, and $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$ be its associated dialectical tree.

Procedure 2.1 (Marking of a dialectical tree). *[8] Let $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$ be a dialectical tree for $\langle \mathcal{A}, h \rangle$. The corresponding marked dialectical tree, denoted $\mathcal{T}^*_{\langle \mathcal{A},\, h \rangle}$, will be obtained marking every node in $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$ as follows:*

1. *All leaves in $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$ are marked as "U"s in $\mathcal{T}^*_{\langle \mathcal{A},\, h \rangle}$.*

[1] To capture the fact that some arguments are stronged *believed* than others, some comparison criteria is needed. In particular, *de.l.p.* uses *specificity* to establish a preference order among arguments [9].

2. *Let $\langle \mathcal{B}, q \rangle$ be an inner node of $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$. Then $\langle \mathcal{B}, q \rangle$ will be marked as "U" in $\mathcal{T}^*_{\langle \mathcal{A},\, h \rangle}$ iff every child of $\langle \mathcal{B}, q \rangle$ is marked as "D". The node $\langle \mathcal{B}, q \rangle$ will be marked as "D" in $\mathcal{T}^*_{\langle \mathcal{A},\, h \rangle}$ iff it has at least a child marked as "U".*

An argument $\mathcal{A}$ is a justification for its conclusion h only if the root in $\mathcal{T}_{\langle \mathcal{A},\, h \rangle}$ is marked as *undefeated*. In the next section we define the concept of agents that use this formalism to represent strict and defeasible knowledge.

DeLP-based agents

As mentioned before, we can use *de.l.p.* to represent the agent's knowledge base, and thus every agent will have the ability to handle inconsistent information. The knowledge base of an agent Ag_i is formed by *extended* and *defeasible program clauses* representing facts and beliefs, and some comparison criteria θ_i to establish a preference order among arguments. Here is a simple definition of DeLP-based agents.

Definition 2.5 (DeLP-based Agent). *A DeLP-based agent $\mathcal{A}_i$ is defined as the pair $(\mathcal{KB}_i, \theta_i)$ such that:*

- *$\mathcal{KB}_i = \{\mathcal{S}_i, \mathcal{D}_i, \}$ where $\mathcal{S}_i$ is a set of strict rules , $\mathcal{D}_i$ is a set of defeasible rules.*
- *$\theta_i : 2^{|\mathcal{D}_i|} \times 2^{|\mathcal{D}_i|} \rightarrow 2^{|\mathcal{D}_i|}$ denotes a preference relation over arguments constructed on $\mathcal{KB}_i$*

For simplicity, we will assume that all the agents in the dialogue use the same comparison criteria, such as *specificity*. The selected criteria is not relevant to this work. The *de.l.p.* formalism can be used to model the agent's internal process of reasoning (monologues). However, the idea of an analysis between arguments and counterarguments can be extended to capture conversational activities between agents. In the next section, we define the concept of *dialogue* in a multi-agent environment together with a simple procedure to engage this kind of social interaction.

3 Dialogues on multi-agent systems

Simply put, a dialogue is sequence of locutionary acts between two or more players. An argument is a tentative explanation for some proposition and when enunciated by agents it may be considered as a locutionary act. Usually, argumentation appears as a mechanism to deal with disagreement between agents, for example, when some conflict of interest is present. In order to define this special kind of social interaction we need a simple definition of what a multiagent system is.

Definition 3.1 (DeLP-based Multiagent system). *A multiagent system (*MAS*) is a set $MS = \{A_1, A_2, A_3, ..., A_n\}$ where every A_i is a DeLP-based agent.*

DeLP-based agents use arguments as an explanation of their beliefs. The process in which the parties (in this case, agents) exchange arguments for or against some proposition is, in fact, a conversational activity between agents and can be depicted as a dialogue.

Definition 3.2 (Dialogue). *An argument dialogue in a multi-agent system MS is a non-empty sequence of pairs*

$$[(Arg_0, Ag_0), (Arg_1, Ag_1), ..., (Arg_i, Ag_j)] \ (i \ \geq \ 0)(1 \ \leq \ j \ \leq \ n)$$

where Arg_i is an argument structure of agent $Ag_j \in MS$. Any pair (Arg, Ag_k) is called a dialogue act *of the dialogue.*

This is the basic definition of dialogue and it can be applied to several forms of social dialogue such as *negotiation*, where there is a conflict of interests, *persuasion* where there is a conflict of opinion or beliefs, *indagation* where there is a need for an explanation or proof of some proposition, *deliberation or coordination* where there is a need to coordinate goals and actions [6]. Other restrictions such as the order in which the agents produce arguments in the dialogue or special conditions for dialogue acts depend on the special kind of dialogue and are given by the *dialogue protocol.* We will give a simple definition of persuasion dialogue, where two agents are arguing about some particular issue. In this kind of dialogue, the agents take turns to present arguments, and what is supplied by each participant at each turn is a direct response to what was stated in the previous turn.

Definition 3.3 (Persuasion Dialogue). *A persuasion dialogue is a dialogue such that for all consecutive dialogue acts (Arg_i, Ag_m) and $(Arg_{i+1}, Ag_n)(i \ \geq \ 0)$ the argument Arg_{i+1} is a defeater argument of Arg_i.*

We also say that Arg_{i+1} is an *answer* or a *response* to Arg_i. In particular, the process of *warranting* an argument in *de.l.p.* as defined in previous sections can be viewed as a set of single-agent persuasion dialogues, defined sometimes as *monologue.* It is easy to see that for an agent A every argumentation line of an argument Arg is a persuasion dialogue in which A is the only agent who makes dialogue acts. This means that for an argument Arg it is possible to make several persuasion dialogues.

Definition 3.4 (Subject of dialogue). *Let $\mathcal{D} = \{(Arg_0, Ag_0), ..., (Arg_i, Ag_j)\}$ be a persuasion dialogue. The first argument Arg_0 is called the* subject *of the dialogue.*

We can think of justification of arguments in terms of a dialogue game between two players P and C. P makes the first argument we are interested in and its defenders and player C makes defeaters. This is shown in [2] for *de.l.p.* and in [3] under another argumentation framework.

Sometimes, when dialoguing, agents find more than one answer to the last *dialogue act* in the sequence. However, as always, an agent must *think* about what to say: only one argument will be selected to participate in the debate. The decision made takes the actual dialogue in one specific direction, perhaps not the best for that agent, as we'll see in examples in the next sections. However, the fact of finding more than one possible response in the debate means that, at some specific time, the agent may *retract* older arguments to take the dialogue in a different course.

3.1 Retracting positions

When involved in a persuasion dialogue $\mathcal{D}$ as defined in 3.3, a DeLP-based agent always introduces defeater arguments for the last argument shown by the opponent in $\mathcal{D}$. As stated before, the concept of persuasion dialogue is similar to the concept of argumentation lines: the agents exchange arguments for and against some proposition (the one supported by the *subject* in the dialogue). For the last argument introduced in the dialogue, an agent A may find more than one defeater. However, only one of them must be selected to be a part of the debate.

Definition 3.5 (Choice Point). *Let $PD = \{(Arg_0, A), ..., (Arg_k, B)\}$ be a persuasion dialogue between two agents A and B. If A finds more than one defeater argument for Arg_k, then (Arg_k, B) is called a* choice *point for A.*

Choice points are very important because they denote the possibility for an agent to present new defeaters for previous arguments in the dialogue. At every choice point, the discussion can flow in different directions, depending on the selections made by the corresponding agent.

Example 3.1. *Let A and B be two agents and let $\mathcal{D}$ be a current dialogue between A and B such that*

$$\mathcal{D} = \{(Arg_1, A), (Arg_2, B)\}$$

Agent A needs to defeat Arg_2, and he finds two defeaters: Arg_{31} and Arg_{32}. Therefore, (Arg_2, B) is a choice point for A. Due to an internal decision process, A decides to present Arg_{31} as a defeater argument for Arg_2.

$$\mathcal{D} = \{(Arg_1, A), (Arg_2, B), (Arg_{31}, A)\}$$

Agent B finds only one defeater argument for Arg_{31}, say Arg_4 and it shows that argument in the dialogue.

$$\mathcal{D} = \{(Arg_1, A), (Arg_2, B), (Arg_{31}, A), (Arg_4, B)\}$$

Now suppose agent A is not able to find any defeater for Arg_4, so A is the loser in the current state of the dialogue. However, A may start a new line of discussion about Arg_2, since A found more than one defeater for that argument. In this case, A retracts Arg_{31} and proposes Arg_{32} as a new defeater for Arg_2. The resulting dialogue is

$$\mathcal{D} = \{(Arg_1, A), (Arg_2, B), (Arg_{32}, A)\}$$

At this point, if B is not able to find defeaters for Arg_{32}, then A is the winner of the dispute.

Every choice point is the subject of several possible dialogues. It is easy to see that these steps in the inter-agent conversation resembles the process of constructing a dialectical tree in the internal process of argumentation. The dialogue ends when an agent is not able to introduce new arguments nor retract older ones.

The next simple algorithms shows how an agent manages the process of debate with other agents. In order to do this, several data structures are constructed: a shared stack $\mathcal{D}$ representing the dialogue, and private stacks C_p for every agent to record choice points. We define an initialization procedure called *Init-Dialogue* an a co-routine *Dialogue* for every agent.

```
Procedure Init-Dialogue
input:  a literal $L$
 Let $A_S$ = { $\langle A_i, L\rangle$:  such that $A_i$ is an argument for $L$}
 Select one argument structure $\langle A_k, L\rangle$ from $A_S$
 Push $\langle A_k, L\rangle$ on $\mathcal{D}$
 Push [$\langle A_k, L\rangle$, $A_S$ - $\langle A_k, L\rangle$ ] on $C_p$
 Give the turn to the other agent.
end Init-Dialogue.
```

```
Co-routine Dialogue
modifies:  $\mathcal{D}$ and $C_p$
 Let $B$ = top($\mathcal{D}$)
 If $B$ is not accepted (due to fallacies or other problem)
   then refuse $B$ (may be poping $B$ from $\mathcal{D}$)
 else
 Let $A_S$ = { $\langle A_i, L\rangle$ such that $\langle A_i, L\rangle$ is a defeater for $B$ }
 if $A_S$ is a singleton
   then push the element of $A_S$ on $\mathcal{D}$
 elsif |$A_S$| > 1
   then
      Select one $\langle A, L\rangle$ from $A_S$
      Push $\langle A, L\rangle$ on $\mathcal{D}$
      Push [$\langle A_k, L\rangle$, $A_S$ - $\langle A_k, L\rangle$ ] on $C_p$
 elsif A = $\emptyset$
   if empty($C_p$) then resign the dialogue
   else /* backtrack to a choice point */
      Pop [$\langle A, L\rangle$, $S$ ] from $C_p$
      Select one defeater $C$ from $S$
      Pop all arguments on $\langle A, L\rangle$ from $\mathcal{D}$ and replace $\langle A, L\rangle$ with $C$
      If $S - \{C\}$ is not empty
      then Push [$\langle A, L\rangle$, $S$ - $\{C\}$ ] on $C_p$
 Give the turn to the other agent.
 end Co-Routine.
```

When an agent introduces an argument, then it pushes that argument onto the stack $\mathcal{D}$, the internal representation of the dialogue. Whenever he finds more than one defeater, he records the corresponding choice point in C_p. This algorithm is used every time the agent receives an argument from the other party in the dialogue. Note that one agent could pop arguments from $\mathcal{D}$ that are in the C_p stack of the other agent. So the selection of one choice point from C_p should consider this possibility.

Definition 3.6 (Winner of the dispute). *An agent A* wins *the persuasion dialogue D against an agent B, if B can not produce a response to any dialogue act of A in D.*

That is, agent B is not able to continue the current dialogue, nor use choice points. Simply put, agent B must resign.

It is important to note that the agent is free to use any choice point or not. Each agent can engage any dialog exhausting all its choice points and also *resign* the dialogue at any time. Of course, in this case the agent loses the dialogue. The guidelines of how and when to take this kind of decisions depends on the communication protocol that is decided to use. Several protocols can be used to engage persuasion dialogues. A good example is explained in [5].

In the next section, we show a new form of constructing a defeater argument to be presented in the dialogue. It is not the intention of this paper to deepen in the subject but, at the moment, to indicate some guidelines of how this task may be achieved.

4 Using the opponent's knowledge

As stated before in this paper, we are interested in dialogues between two agents. Persuasion dialogues for DeLP-based agents are basically handled as follows: an agent A proposes an argument Arg_1 to the agent B. Agent B now must present a defeater argument for Arg_1. In order to do this, B looks for all the possible counterarguments of Arg_1 in its knowledge base KB_A, and starts a dialectical analysis for all of them, trying to find a *warranted* defeater, say Arg_2. The agent B then proposes Arg_2 as a defeater of Arg_1, and it is now A's turn to produce a defeater for Arg_2, and so on. This process is shown in [5] applied to a different framework.

What we propose in this section is a new technique to build defeater arguments in a persuasion dialogue between two *DeLP-based agents*. It uses not only the knowledge base of the agent, but also any information shown in the actual dialogue. The next example depicts a situation in which a dialogue can be engaged using part of opponent's beliefs.

4.1 Example

In order to refute the last argument in the dialogue, it is necessary to find defeater arguments. Usually, when an agent needs to construct an argument, this task is based in two important items: its knowledge base and the actual state of the dialogue. The former is used to construct arguments, the latter is used to avoid circular argumentation or contradiction as in [9]. When two agents are involved in a dialogue, they exchange arguments reflecting part of each agent's beliefs.

Example 4.1. *Let A and B be two DeLP-based agents with the following knowledge bases.*

$$S_1 = \{r; w; v\}$$
$$D_1 = \{(p \prec q, w)(q \prec r)(p \prec v)\}^2$$

$$S_2 = \{t; r\}$$
$$D_2 = \{(\sim p \prec s)(s \prec t)(\sim p \prec q)\}$$

In this case, agent A may construct the argument $\langle Arg_1, p\rangle$ where

[2] Parentheses are used to distinguish one rule from another

$$Arg_1 = \{(p \prec q, w)(q \prec r)\}$$

The agent B is able to respond with the defeater $\langle Arg_2, \sim p\rangle$ *where*

$$Arg_2 = \{(\sim p \prec s)(s \prec t)\}$$

The agent A now must construct a defeater for Arg_2*. A can not present* Arg_1 *as a defeater in order to avoid circular argumentation. However, A is able to produce the argument* $\langle Arg_3, p\rangle$*, a new defeater for* Arg_2 *where*

$$Arg_3 = \{p \prec v\}$$

Now, agent B can not construct new arguments according to its own knowledge base. However, the defeasible rule $q \prec r$ *presented by agent B will be useful to produce the defeater argument* Arg_4 *for* Arg_3 *where*

$$Arg_4 = \{(\sim p \prec q)(q \prec r)\}$$

If this is possible, then B is the winner of the dispute, because A is not able to produce new defeaters for Arg_4*.*

In this example, agent B has used one defeasible rule of A's knowledge base to make a refutation of A's proposal. The new rules do not affect the internal knowledge of the agent. They are used solely with a temporary objective. By observing the opponent's arguments, sometimes it is possible to produce an argument partially based on external beliefs. This procedure allows the agent to continue with the dialogue, avoiding imminent resignations. Even more, using opponent's rules minimizes the possibility of future refutations on that argument.

4.2 Selecting opponent's rules

To borrow some rules from the opponent in order to produce new arguments in the dialogue is not an easy task. The agent must be careful in not falling into contradiction with itself. This is so because the new rules can interfere with the beliefs of the agent. The next definition is the starting point to analyze the rules that can be *borrowed* by an agent.

Definition 4.1. *Let* $\mathcal{D}$ *be a persuasion dialogue between two DeLP-based agents. The set*

$$Info_{\mathcal{D}} = \{r : r \in Arg \textit{ for all } Arg \in \mathcal{D}\}$$

is the set of all the information concerning to the dialogue.

The set $Info_{\mathcal{D}}$ is a set of defeasible rules. A DeLP-based agent trying to refute some argument Arg_i is able to use $Info_{\mathcal{D}}$ as an additional information source in order to produce counterarguments for Arg_i. The agent may select one or more rules of $Info_{\mathcal{D}}$ and then present a new argument in the dialogue. Note that in this set there are rules of both dialoguing agents. We will focus in the scenario where an agent A tries to use rules of $Info_{\mathcal{D}} - KB_A$ in search of new future responses in the debate.

Definition 4.2. *A defeasible rule r is called a* concordant rule *for the set of rules $\mathcal{R}$ if for any argument Arg that is a justified argument in $\mathcal{R}$ then Arg is also a justified argument in $\mathcal{R} \cup \{r\}$. A set $\mathcal{S}$ of defeasible rules is a* concordant set *for $\mathcal{R}$ if for all argument Arg that is a justified argument in $\mathcal{R}$ then A is also a justified argument in $\mathcal{R} \cup \mathcal{S}$.*

In any concordant set $\mathcal{S}$ for $\mathcal{R}$, then every rule $r \in \mathcal{S}$ is a concordant rule for $\mathcal{S} \cup \mathcal{R}$ For a DeLP-based agent A in a dialogue $\mathcal{D}$, it is interesting to construct arguments from its knowledge base KB_A and some concordant set D for KB_A such that $D \subseteq Info_{\mathcal{D}} - KB_A$.

Definition 4.3. *A defeasible rule r is called a* controversial rule *for the set of rules $\mathcal{R}$ if there exists an argument Arg that is a justified argument in $\mathcal{R}$ but not in $\mathcal{R} \cup \{r\}$. A set $\mathcal{S}$ of defeasible rules is a* controversial set *for $\mathcal{R}$ if there exists an argument Arg that is a justified argument in $\mathcal{R}$ but not in $\mathcal{R} \cup \mathcal{S}$.*

An agent A cannot use a controversial rule to construct arguments for the dialogue, because that rule is in opposition to what A believes.

Example 4.2. *Let A and B be two DeLP-based agents with the following knowledge bases.*

$$S_1 = \{r, s\}\ ,\ D_1 = \{(q \mathrel{-\!\!\prec} r)(p \mathrel{-\!\!\prec} q)({\sim} p \mathrel{-\!\!\prec} w)\}$$

$$S_2 = \{r\}\ ,\ D_2 = \{({\sim} p \mathrel{-\!\!\prec} w)(w \mathrel{-\!\!\prec} r)\}$$

Agent A cannot use the rule $w \mathrel{-\!\!\prec} r$ from B's knowledge base because it allows to build the argument $\langle\{({\sim} p \mathrel{-\!\!\prec} w)(w \mathrel{-\!\!\prec} r)\}, {\sim} p\rangle$ which is a defeater of $\langle\{q \mathrel{-\!\!\prec} r)(p \mathrel{-\!\!\prec} q)\}, p\rangle$

Note that if we add a controversial rule to any concordant set then we not necessarily get a controversial set.

Definition 4.4. *A defeasible rule r_1 is called a* reinstating rule *for the set of rules $\mathcal{R}$ if for all argument Arg and a controversial rule r_2 , Arg is a justified argument in $\mathcal{R} \cup \{r_1, r_2\}$.*

That is, if we add r_2 to the set $\mathcal{R} \cup \{r_1\}$ then the set of justified arguments in the union is equal to the set of justified arguments in R.

Example 4.3. *In example 4.2, if B has the rule $p \mathrel{-\!\!\prec} s$ as part of its knowledge base, then A may use it together with the controversial rule $w \mathrel{-\!\!\prec} r$, because it produces the argument $\langle\{p \mathrel{-\!\!\prec} s\}, p\rangle$ which defeats $\langle\{({\sim} p \mathrel{-\!\!\prec} w)(w \mathrel{-\!\!\prec} r)\}, {\sim} p\rangle$ reinstating $\langle\{q \mathrel{-\!\!\prec} r)(p \mathrel{-\!\!\prec} q)\}, p\rangle$.*

An agent A that is dialoguing with another agent B is able to select any rule that takes part in some argument shown by B, only if that rule is not a controversial rule for KB_A. This is the case of the example in the previous section.

Of course, it's desirable sometimes to borrow more than one rule from the opponent. This is a much more complex situation because the union of concordant sets is not necessarily a concordant set. What is needed here in a dialogue $\mathcal{D}$ is to find a concordant set for KB_A in $Info_{\mathcal{D}}$. However, there may exist several concordant subsets $\{CS_1, CS_2, ..., CS_k\}$ in $KB_A - Info_{\mathcal{D}}2$ for KB_A. The agent should not select rules from one set CS_i and later use rules of another concordant set CS_j because of the inherent risk of falling into contradiction: the union of these sets may not be a concordant set for KB_A.

Definition 4.5. *A set $\mathcal{S}$ of defeasible rules is a* maximal concordant set *for $\mathcal{R}$ if there not exists $\mathcal{S}' \supseteq \mathcal{S}$ such that $\mathcal{S}'$ is a concordant set for $\mathcal{R}$.*

Maximal concordant sets group the rules that can be considered in the attempt to construct new arguments. Again, it is possible to find more than one maximal concordant set.

It is important to note that in this scenario there is not any process of belief revision. The rules are borrowed only in order to produce an alternative refutation to the last argument in the dialogue. It is only an alternative to engage the dialogue. The agent is not forced to choose rules, but he must observe these properties and only use concordant rules for its knowledge base.

5 General architecture

An agent A involved in a dialogue with another agent B needs to produce arguments to refute the last argument of the opponent. In order to do this, A must use its knowledge base KB_A and it's able to use some rules shown in the dialogue by agent B. The general outline of any dialoguing DeLP-based agent is shown in Figure 1 where the arrows represent information flow.

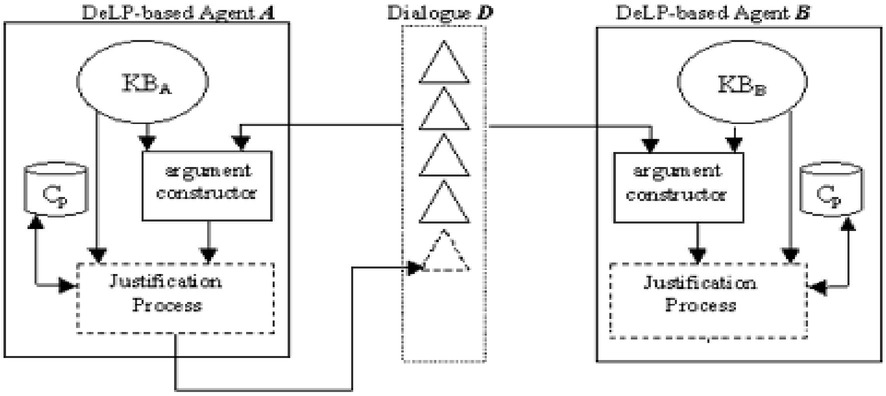

Figure 1: Dialoguing agents

An agent may have an internal representation of the dialogue that is being carried out, although in the figure they are drawn separated. The inputs of the argument constructor are the local knowledge base and $Info_{\mathcal{D}}$, that is, the set of rules shown in the dialogue. This unit may produce an argument Arg, but in order to present Arg in the debate, this argument must be analyzed with the information in the knowledge base, through a complete dialectic process. Thus, only locally justified arguments are introduced in the dialogue.

6 Conclusions and future work

A multi-agent system is made up of multiple interacting autonomous agents cooperating to achieve common goals, or competing for them. They establish dialogues via some kind of agent-communication language, under some communication protocol. We think argumentation is suitable to model several kind of dialogues in multi-agents systems. Two main contributions are presented in this paper. First, we defined the concept of DeLP-based agents. These agents use Defeasible Logic Programs [9, 8] as a knowledge representation language. We defined dialogues and persuasion dialogues between this kind of agents and we presented an algorithm to establish how this dialogue may be engaged. Second, a brief description of how an agent could use opponent's information for its own benefit is given. This is useful to avoid imminent resignations and minimizes the possibility of refutations on the arguments. In the future, we expect to define some procedure to select the best set of concordant rules for the knowledge base of an agent, and analyze the role of the argument comparison criteria of every agent in the dialogue.

Bibliografía

[1] Diego Martínez y Alejandro García. *Social interaction and argumentation in MAS.* Proceedings of IV WICC, 2001.

[2] Laura Cecchi, Guillermo Simari. *Una semántica declarativa basada en juegos para la programación en lógica rebatible básica.* Proceedings of ICIE, 2000.

[3] Leila Amgoud, Simon Parsons and Nicolas Maudet. *Arguments, dialogue and negotiation* Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, 2000.

[4] Phan Minh Dung, *On the Acceptability of Arguments and its Fundamental Role in Nonmotonic reasoning and Logic Programming and N-persons games*, IJCAI 93 852-857, 1993.

[5] Aleandro Stankevicius y Guillermo Simari *A Framework for Multiagent Deliberation Based on Dialectical Argumentation Proceedings of CACIC 2001.*

[6] Douglas Walton. *The New Dialectic: A Method for Evaluating an Argument used Purpose in a Given Case. Proto Sociology: An International Journal of Interdisciplinary Research, Vol. 13, 1999. pp 70-91.*

[7] G. Weiss. *Multiagents Systems: A Modern Approach to Distributed Artificial Intelligence.* G. Weiss, editor. MIT Press. Cambridge, Massachussets. 1999.

[8] Alejandro J. García and Guillermo R. Simari. *Defeasible Logic Programming: An Argumentative Approach.* Theory and Practice of Logic Programming, 2002. Accepted for publication.

[9] Alejandro J. García. *Defeasible Logic Programming: Definition, Operational Semantics and Parallelism.* PhD Thesis. Computer Science Department, Universidad Nacional del Sur, Bahía Blanca, Argentina.

[10] Alejandro J. García and Guillermo R. Simari *Parallel Defeasible Argumentation.* Journal of Computer Science and Technology Special Issue: Artificial Intelligence and Evolutive Computation. Vol 1, No. 2, 45–57, 1999. http://journal.info.unlp.edu.ar/