# Cooperative Dialogues with Conditional Arguments

Samy Sá
Universidade Federal do Ceará
Campus do Pici, Bl 910
Fortaleza, Brazil
samy@ufc.br

João Alcântara
Universidade Federal do Ceará
Campus do Pici, Bl 910
Fortaleza, Brazil
jnando@lia.ufc.br

## ABSTRACT

We introduce an approach to cooperative dialogues as a framework for group deliberation. One of its distinguishing features is that it deals with conditional and constraint-based arguments, which are built by employing abductive and hypothetical reasoning. These kinds of arguments allow agents to use a variety of dialogue moves proper to a cooperative debate, such as argument rewrites and conditional attacks. In our approach, a group of agents develops a dialogue as they explore different lines of thought to build a group position in a yes or no decision. In essence, given a matter for discussion, the parties involved will consider arguments that either supports or rejects it and discuss such arguments to decide whether or not to accept them. To achieve that, agents will work as a team and combine their knowledge to produce more complex arguments and study possible flaws these might have.

## Categories and Subject Descriptors

F.4.1 [**Mathematical Logic**]: Logic and constraint programming; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

## General Terms

Theory

## Keywords

Collective Intelligence, Reasoning (single and multiagent), Argumentation, Logic Programming, Abduction

## 1. INTRODUCTION

Dialogues were introduced into multiagent systems to formalize the generation and interpretation of arguments exchanged amongst agents [1]. Such dialogues are perceived as a game involving two antagonistic agents in a discussion about some matter (a proposition). In this setting, the first player tries to justify or defend the matter, while the second will try to disqualify or attack it. However, we consider that some dialogues are inherently cooperative in the sense that the agents in a group might share a goal and be interested in working together to justify or disqualify a proposition. This is the case with Deliberation Dialogues [14], which is our main focus. We believe that the key for this kind of cooperation lies in abductive reasoning [7], which is a special kind of non-monotonic reasoning, usually defined as inference to the best explanation.

In this paper we focus on deliberation dialogues about whether the group can explain a scenario or if they should accept an argument from an external source. Our goal is to allow agents to collectively engage in reasoning about what to do, however without the need to share their entire knowledge bases. This is an important feature, since in human-agent interaction, knowledge bases cannot be simply merged. We can also think of a self-preservation rationality, for an agent might experience disadvantages if it later gets into a negotiation or game involving an agent with whom it just shared its entire knowledge base. On the other hand, the kind of deliberation we propose is hardly as efficient as joining the knowledge bases to draw collective conclusions.

In existing approaches to dialogues in multiagent systems [1, 14], the agents in a group will share opinions (arguments) to reach a consensus on which ones are good. Their knowledge, however, is combined in a very limited way because every opinion is proposed by a single agent. In our work, agents engage in hypothetical reasoning to consider alternative scenarios and combine their knowledge further. As a consequence, agents can cooperate to complement the arguments from one another and reach a deeper understanding of the possible flaws their arguments might have.

Our work defines a framework for the exchange of arguments between agents, such as in multiagent dialogues [1] and negotiation with abduction [17]. Multiagent dialogues are characterized in [1] as a game involving two agents in antagonistic positions about some matter of discussion, possibly a deal in a process of negotiation. Agents will place arguments attacking each other opinions until one of them can no longer respond, so the other will be the winner. Similarly, negotiation is perceived as an exchange of proposals, and whenever an agent can no longer counter the last, it has to accept it. Negotiation was improved with abduction in [17], where the authors introduce the possibility of conditional proposals based on abductive reasoning. In cooperative deliberation as we introduce in this paper, agents work as team mates and explore alternatives by exchange arguments for the best interest of the group.

Amongst others, our framework has the following characteristics: $(i)$ agents can resort to hypothetical reasoning to produce arguments; $(ii)$ the agents in a group might be

**Appears in:** *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

able to combine their knowledge to produce elaborate arguments that no single agent can conceive on its own; (*iii*) the reasoning performed by a group of agents involves two opposing consistent positions; (*iv*) the dialogues are guaranteed to end, so the agents are sure to reach an agreement about the subject of discussion.

In Section 2, we will present abductive logic programs as they are used throughout the paper. Next, we will introduce conditional arguments in Section 3 and our approach to collective dialogues in Section 4. These last two sections hold our main contributions. We discuss related work in Section 5 and conclude the paper in Section 6 with a discussion on the importance of our contributions and future work.

# 2. PRELIMINARIES

## 2.1 Extended Disjunctive Programs

In this paper, we account for programs as in Extended Disjunctive Programs (EDP's) [10] without disjunctive heads.

An EDP is defined over a *Herbrand Universe* $HB$, the set of all ground atoms the program might resort to. Such a program consists of a set of rules of the form

$$r:\ L_H \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$$

with $L_H$ being optional and $n \geq m \geq 0$. In this notation, each $L_i$ is a literal (an atom $A$ or its negation $\neg A$), $L_H$ is a literal, and *not* is *negation as failure* (NAF). If $L$ is a literal, $not\ L$ is called a NAF-*Literal*. We might speak of literals to generalize literals and NAF-Literals. In a rule $r$ on the above form, we refer to $L_H$ as the *head* of the rule and write $head(r)$ to denote the set $\{L_H\}$. We refer to the conjunction $L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$ as the body of $r$, and $body(r)$ denotes the set $\{L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n\}$. We differ the literals of its positive and negative parts as $body^+(r)$ and $body^-(r)$ to refer to the sets $\{L_1, \ldots, L_m\}$ and $\{L_{m+1}, \ldots, L_n\}$, respectively. We also denote $not\_body^-(r)$ as the set of NAF-*Literals* $\{not\ L_{m+1}, \ldots, not\ L_n\}$. A rule may be written as $head(r) \leftarrow body^+(r), not\_body^-(r)$ or $head(r) \leftarrow body(r)$, for $body(r) = body^+(r) \cup\ not\_body^-(r)$. A rule is an integrity constraint if $head(r) = \emptyset$ and it is a fact if $body(r) = \emptyset$, in which case we do not write "$\leftarrow$". We say a program is NAF-free if it does not contain NAF-Literals.

The semantics of an EDP is given by the Answer Sets Semantics [10]. Consider $Lit_P$ is the set of all literals in the language of a program $P$ and $S$ one of its subsets. Let $P^S$ be the set that contains all the instances $head(r) \leftarrow body^+(r)$ of rules of $P$ such that $body^-(r) \cap S = \emptyset$ and no other rules, so $P^S$ is a NAF-free program. Given a NAF-*free* EDP $P$, $Ans(P)$ is a minimal subset of $Lit_P$ such that (*i*) for every ground rule of $P$, if $body^+(r) \subseteq S$, then $head(r) \in S$ and (*ii*) S is either consistent or $S = Lit_P$. Given an EDP $P$, $S$ will be an answer set of $P$ if $S = Ans(P^S)$. A program might have zero, one or multiple answer sets. An answer set $S$ for $P$ is consistent if $S$ does not simultaneously contain $A$ and $\neg A$, for no atom in the language. The program itself will be said consistent if it has a consistent answer set. Otherwise, the program is inconsistent.

We draw special attention to the following terminology:

- A *goal* is a conjunction of literals and NAF-literals. If $G$ is a goal, then $Lit(G)$ is the set of literals and NAF-literals in $G$. If $Hyp$ is a set of rules (a program), $Lit(Hyp) = \{L \in (body(r) \cup head(r)) |\ r \in Hyp\}$.
- An EDP $P$ satisfies $G = L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$ (written $P \models G$) if $P$ has an answer set $S$ such that $\{L_1, \ldots, L_m\} \subseteq S$ and $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$.
- Given a literal or NAF-literal $L$, we have: if $L = A$, then $neg(L) = \neg A$; if $L = \neg A$, then $neg(L) = A$; if $L = not\ L'$, $L' \in \{A, \neg A\}$, then $neg(L) = L'$.

## 2.2 Abductive Logic Programs

Abduction is a special kind of non-deductive reasoning in which hypotheses are inferred to explain observable facts otherwise not accepted by a theory. Abductive Logic Programming brings this feature to standard logic programming [12, 7]. We will now introduce Abductive Logic Programs (ALP's) as in the abductive framework of *Extended Abduction* [16, 17], but adapted to our objectives.

An abductive program is a pair $\langle P, H\rangle$, where $P$ is an EDP and $H$ is a set of literals referred to as abducibles. If $P$ is consistent, then $\langle P, H\rangle$ is consistent. Throughout the paper we will assume only consistent programs. A goal[1] is satisfied by $\langle P, H\rangle$ if $\{L_1, \ldots, L_m\} \subseteq S$ and $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$ for some answer set $S$ of $P$.

Definition 1. *Let $G$ be a goal for the ALP $\langle P, H\rangle$. A pair $(E, F)$ is an* explanation *to $G$ in $\langle P, H\rangle$ if*

1. *$(P \setminus F) \cup E$ has an answer set which satisfies $G$*[2]*,*
2. *$(P \setminus F) \cup E$ is consistent,*
3. *$E$ and $F$ are sets of literals such that $E \subseteq H \setminus P$ and $F \subseteq H \cap P$.*

Intuitively, an explanation $(E, F)$ of $G$ in $\langle P, H\rangle$ means that by assuming the literals in $E$ as true while retracting (falsifying) the literals in $F$ from $P$, the resulting $P' = (P \setminus F) \cup E$ satisfies $G$. If the original program has an answer set satisfying $G$, then $(\emptyset, \emptyset)$ is an explanation and no changes are needed in $P$. An explanation $(E, F)$ is minimal if, for any explanation $(E', F')$ such that $E' \subseteq E$ and $F' \subseteq F$, then $E' = E$ and $F' = F$. In general, only the minimal explanations are of interest.

If an agent has a program $\langle P, H\rangle$ as its knowledge base, the set $H$ lists the literals the agent can resort to for hypothetical reasoning. These literals might not be enough to explain some goals, so an agent should be able to adapt and consider unknown literals from such goals as abducibles too. This capability might not be helpful when the agent is on its own, but it can be essential in a group deliberation setting.

Definition 2. *Given a goal $G$ and a program $\langle P, H\rangle$, $Ab(P, H, G)$ is the set of literals that appear in $G$ but appear neither in $P$ nor in $H$. An agent adapts to deliberate on $G$ if it considers the literals in $Ab(P, H, G)$ as abducibles, i.e., if it reasons as if its knowledge base were $\langle P, H \cup Ab(P, H, G)\rangle$.*

Example 1. *Consider the following ALPs:*

| | | | | | |
|---|---|---|---|---|---|
| $P_1:$ | $a \leftarrow b;$<br>$b \leftarrow c;$ | $P_2:$ | $b \leftarrow d;$<br>$d \leftarrow not\ e;$<br>$e$ | $P_3:$ | $\leftarrow c, d;$<br>$d;$ |
| $H_1:$ | $\{b, c\}$ | $H_2:$ | $\{d, e\}$ | $H_3:$ | $\{d, c\}$ |

[1]In [17], goal is referred as observation.

[2]This definition is for *credulous* explanations. Its choice over *skeptical* explanations [11] makes possible to have more explanations and gives us a better chance of finding good related arguments in a discussion.

*Consider a group of three agents $Ag_1, Ag_2, Ag_3$. Each agent $Ag_i$ has its knowledge base in ALP $\langle P_i, H_i \rangle$, $i = 1, 2, 3$. These agents can build, amongst others, the following minimal explanations (as in Definition 1):*

$Ag_1: \; Ex_1 = (E_1, F_1) = (\{b\}, \{\ \})$ *for* $G_1 = a, b$.
$Ag_1: \; Ex_2 = (E_2, F_2) = (\{b, d\}, \{\ \})$ *for* $G_2 = a, d, not\ c$.
$Ag_2: \; Ex_3 = (E_3, F_3) = (\{d\}, \{\ \})$ *for* $G_3 = b$.
$Ag_2: \; Ex_4 = (E_4, F_4) = (\{\ \}, \{e\})$ *for* $G_4 = d$.
$Ag_3: \; Ex_5 = (E_5, F_5) = (\{c\}, \{d\})$ *for* $G_5 = c$.

*The first explanation suggests that if b is true, $Ag_1$ can prove $G_1 = a, b$. The second explanation uses the literal $d \in Ab(P_1, H_1, G_2)$, which is not in $P_1$, so the agent has adapted to deliberate on $G_2$ (Definition 2) and has built the explanation $Ex_2$ in $\langle P_1, H_1 \cup Ab(P_1, H_1, G_2) \rangle$. It means that $Ag_1$ cannot justify d in $G_2$ and that if b is true, $Ag_1$ can prove the part of $G_2$ that does not involve d, i.e., $G_2{}' = a, not\ c$. The explanation $(E_3, F_3)$ has a similar meaning to that of $(E_1, F_1)$, and is highlighted because it can be combined with the latter to create a new explanation to $G_1$, as we will later explore in the paper. The next explanation states that $Ag_2$ believes e is true ($\langle P_2, H_2 \rangle \models e$), but is capable to conceive e being false, as it would explain $G_4 = d$. Finally, $Ag_3$ could prove $G_5 = c$ under the conditions that c is true and d is false, but it believes d is true and has no opinion about c. In each case, an explanation is intended to mean that an agent is willing to discuss some of its knowledge.*

*Further, the goal $G_6 = a, b, d$ cannot be satisfied by any of the three programs, even though the program $\langle P_\cup, H_\cup \rangle$, where $P_\cup = \bigcup P_i$ and $H_\cup = \bigcup H_i$, $i = 1, 2, 3$, satisfies it. We highlight that if the agents adapt (Definition 2), they can produce explanations to $G_6$. In the next section, we will show explanations play a key role in the definition of conditional arguments. Also, we will show agents capable of abductive reasoning as above can satisfy $G_6$ in the example as they share and complement each others explanations.*

# 3. CONDITIONAL ARGUMENTS

Intuitively, an argument consists of a conclusion and some justification to it. The reading of an argument is that if its justification is acceptable, the conclusion should be as well. These arguments might be somehow defective, so other arguments can be proposed to point its possible flaws. In this section, we will formalize arguments and proceed to extend this notion with hypothetical reasoning. To that sense, we will introduce two kinds of *conditional* arguments: The first kind is based on hypothetical scenarios in which a particular conclusion makes sense. Using this type of argument can enrich the discussion of a matter, as it allows agents to go deeper on exploring the possible flaws the arguments might have and to cooperate with other agents by combining their knowledge. The second kind of argument involves arguing that some hypotheses from an argument can lead to absurd conclusions, so it should be rejected.

Definition 3 (Arguments). *An argument in an EDP P is a pair $(Hyp, G)$ where G is a goal and Hyp is a set of instances of rules of P such that (i) there is a consistent answer set of P that satisfies Hyp; (ii) $Hyp \models G$ and (iii) Hyp is minimal, so no $Hyp' \subset Hyp$ satisfies both i and ii.*

If $(Hyp, G)$ is an argument, the set $Hyp$ is called the support or hypotheses set and $G$ is the *conclusion* of the argument. Given an argument $Arg = (Hyp, G)$, its set of literals and NAF-literals is $Lit(Arg) = Lit(Hyp) \cup Lit(G)$.

In a dialogue, an agent can disagree with others by attacking their arguments:

Definition 4. *An argument $Arg_1 = (Hyp_1, G_1)$ attacks $Arg_2 = (Hyp_2, G_2)$ if there is a $L_1 \in Lit(G_1)$ such that $L_1 = neg(L_2)$ for some literal $L_2 \in Lit(Arg_2)$.*

Example 2. *Consider an agent Ag with knowledge base represented by the following EDP P:*

$$P: \; a \leftarrow not\ b; \quad c$$

*The goal $G_1 = a, c$ is satisfied by P, so Ag can produce an argument $Arg_1 = (\{a \leftarrow not\ b; c\}, G_1)$ to explain it to the other agents. The goal $G_2 = a, \neg c$, however, is not satisfied by P, and Ag can produce the argument $Arg_2 = (\{c\}, c)$ to suggest it cannot be satisfied by the group, since c denies $\neg c$ in $G_2$. The conclusion of $Arg_3 = (\{\neg c \leftarrow a; a\}, \neg c)$ denies an hypothesis of $Arg_2$, so $Arg_3$ attacks $Arg_2$.*

## 3.1 Abduction-Based Conditional Arguments

An agent capable of abductive reasoning can conceive alternative hypothetical scenarios in which a goal would be satisfied. The agent can then build arguments in any alternative scenario and highlight the conditions in which the scenario would be acceptable. We refer to these arguments as *conditional arguments*, for they can only be accepted by a group of agents if the conditions presented in the argument are satisfied by them. A notion of conditional arguments has been introduced in [13], but the following definitions are original and based on extended abduction (Section 2.2).

Definition 5 (Conditional Arguments). *Consider $\langle P, H \rangle$, an ALP, and $(E, F)$, a minimal explanation to the goal G. The tuple $(E, F, Hyp, G)$ is a* conditional *argument to G if $(Hyp, G)$ is an argument in $P\,' = (P \setminus F) \cup E$. An argument $(E, F, Hyp, G)$ with $E = F = \emptyset$ is* non-conditional.

If $(E, F, Hyp, G)$ is a conditional argument, $Hyp$ is its support or hypothesis set, $G$ is the *conclusion* and each element of $Hyp$ is a hypothesis. To denote the set of conditions that the explanation adds to the argument, we write $C(E, F) = E \cup \{not\ L \mid L \in F\}$. If $Arg = (E, F, Hyp, G)$ is a conditional argument, we denote its set of literals and NAF-Literals as $Lit(Arg) = C(E, F) \cup Lit(Hyp) \cup Lit(G)$.

The idea is that a conditional argument proposed by an agent would be accepted in our framework if the explanation in it is justified by the other agents.

Example 3. *Consider an agent Ag with knowledge base represented by the following ALP $\langle P, H \rangle$:*

$$P: \; a \leftarrow b; \quad c; \qquad H: \; \{b, c\}$$

*The goal $G = a, c$ is not satisfied by P, but Ag can produce the explanation $(E, F) = (\{b\}, \{\})$ and build the conditional argument $A = (\{b\}, \{\ \}, \{b;\ a \leftarrow b;\ c\},\ G)$ in $P\,' = P \cup \{b\}$.*

The definition of attack with conditional arguments is about the same as before, though now it is also possible to attack the explanation attached to an argument.

DEFINITION 6. *An argument* $Arg_1 = (E_1, F_1, Hyp_1, G_1)$ *attacks* $Arg_2 = (E_2, F_2, Hyp_2, G_2)$ *if there is a* $L_1 \in Lit(G_1)$ *such that* $L_1 = neg(L_2)$ *for some literal* $L_2 \in Lit(Arg_2)$. *If* $(E_1, F_1, Hyp_1, G_1)$ *is an attack to some argument and* $C(E_1, F_1) \neq \emptyset$, *we say it is a* conditional attack.

EXAMPLE 4. *Consider* $Arg_1 = (\{b\}, \{\ \}, \{b;\ a \leftarrow b;\ c\},\ G)$, $G = a, c$. *The arguments* $Arg_2 = (\{\ \}, \{\ \}, \{\neg b \leftarrow not\ d;\}, \neg b)$ *and* $Arg_3 = (\{c\}, \{\ \}, \{c;\ \neg b \leftarrow c;\},\ \neg b)$ *are examples of attacks to* $Arg_1$ *that focus on its conditions. On top of that,* $Arg_3$ *is a conditional attack to* $Arg_1$.

## 3.2 Building Arguments Together

A conditional argument should only be accepted by a group of agents if its conditions are satisfied by other agents. The agents in a group should therefore cooperate to reduce the number of conditions of an argument and transform it into a non-conditional argument (Definition 5). This is done by *rewriting* a conditional argument, i.e., adding support to some of its conditions, even though it might be necessary to introduce others.

DEFINITION 7 (ARGUMENT REWRITE). *Consider a conditional argument* $Arg_1 = (E_1, F_1, Hyp_1, G_1)$ *proposed by agent* $Ag_1$. *Also, consider an agent* $Ag_2$ *can produce an argument* $Arg_2 = (E_2, F_2, Hyp_2, G_2)$ *such that* $G_2 = L_1$, *for some* $L_1 \in E_1$. *Then,* $Ag_2$ *rewrites* $Arg_1$ *as* $Arg_1{}' = ((E_1 \setminus \{L_1\}) \cup E_2,\ F_1 \cup F_2,\ (Hyp_1 \setminus \{L_1\}) \cup Hyp_2,\ G_1)$.

In our framework, an argument rewrite will consist of a dialogue move in which an agent attempts to unify its opinion with those of other agents. In particular, if $(E_2 = \emptyset$, the rewriting consists in reducing the cardinality of $E_1$, being therefore an attempt to fulfill the conditions of $Arg_1$.

EXAMPLE 5. *Consider* $Arg_1 = (\{a\}, \{\ \}, \{b \leftarrow a;\ a\},\ b)$ *and* $Arg_2 = (\{\ \}, \{\ \}, \{a \leftarrow c;\ c\},\ a)$. *It is possible to rewrite* $Arg_1$ *into* $Arg_1{}' = (\{\ \}, \{\ \}, \{b \leftarrow a;\ a \leftarrow c;\ c\},\ b)$, *which is a non-conditional argument to* $b$.

*Now consider* $Arg_2{}' = (\{c\}, \{\ \}, \{a \leftarrow c;\ c\},\ a)$. *We can rewrite* $Arg_1$ *as* $Arg_1{}'' = (\{c\}, \{\ \}, \{b \leftarrow a;\ a \leftarrow c;\ c\},\ b)$, *which is a conditional argument with different conditions.*

The process of rewriting allows for agents to cooperate and build more elaborate arguments. In fact, any sequence of argument rewrites $Arg^0, Arg^1, \ldots, Arg^n$, provides a new argument, possibly conditional, that combines the knowledge of as many agents as the number of authors of arguments in that sequence. In particular, if the sequence ends with $Arg^n = (\emptyset, F^n, Hyp^n, G)$, the argument is eligible for being accepted in our framework.

EXAMPLE 6. *Consider the agents and the goal* $G_6 = a, b, d$ *taken from Example 1, together with the explanations below:*

$$Ag_1 :\ Ex_6 = (E_6, F_6) = (\{b, d\}, \{\ \})\ for\ G_6.$$
$$Ag_2 :\ Ex_3 = (E_3, F_3) = (\{d\}, \{\ \})\ for\ G_3 = b.$$

*The agent* $Ag_1$ *cannot satisfy* $G_6$ *but can build the explanation* $Ex_6$ *to it. The agent can build the conditional argument* $A_1 = (\{b, d\}, \{\ \}, \{b;\ a \leftarrow b;\ d\},\ G_6)$ *in* $P' = P \cup \{b, d\}$. *Then,* $Ag_2$ *should try to fulfill the conditions of* $A_1$, *with* $A_2 = (\{d\}, \{\ \}, \{d;\ b \leftarrow d;\},\ G_3)$ *to rewrite* $Arg_1$ *into* $Arg_1{}' = (\{d\}, \{\ \}, \{d;\ b \leftarrow d;\ a \leftarrow b;\},\ G_6)$, *which has less conditions. Finally,* $Ag_3$ *can complement this condition, since* $P_3 \models d$. *The third agent uses* $A_3 = (\{\ \}, \{\ \}, \{d\},\ d)$ *to rewrite* $Arg_1{}'$ *into* $Arg_1{}'' = (\{\ \}, \{\ \}, \{d;\ b \leftarrow d;\ a \leftarrow b;\},\ G_6)$, *which is a non-conditional argument.*

Arguments built with rewrites are possibly not derivable by any of the agents individually (only by the group), which is the case with $Arg_1{}''$ above. An important property of such arguments is that any attacks the agents can place against $Arg_i$, will also attack those obtained by rewriting $Arg_i$.

THEOREM 1. *If* $Arg_{i+1} = (E_{i+1}, F_{i+1}, Hyp_{i+1}, G_{i+1})$ *is a rewrite of* $Arg_i = (E_i, F_i, Hyp_i, G_i)$ *and there is an argument* $Arg_j = (E_j, F_j, Hyp_j, G_j)$ *that attacks* $Arg_i$, *then* $Arg_j$ *is also an attack to* $Arg_{i+1}$.

PROOF. By exhaustion, suppose $Arg_j$ is an attack that negates a literal or NAF-literal $L$ in

$G_i$: Since $G_{i+1} = G_i$, $Arg_j$ also attacks $Arg_{i+1}$;

$Hyp_i$: Observe that the process of rewriting arguments will never remove hypotheses of $Hyp_i$ that are program rules, except for facts. Because $Hyp_i$ is minimal, there should be at least one rule $r \in Hyp_i$ with $L \in body(r)$. As this rule is still a hypothesis in $Arg_{i+1}$, we conclude that $Arg_j$ is an attack to $Arg_{i+1}$;

$C(E_i, F_i)$: Given that $(E_i, F_i)$ is a minimal explanation, each and every condition is also in the body of at least one rule $r \in Hyp_i$ or in the goal $G_i$. Therefore, even if $L \notin C(E_{i+1}, F_{i+1})$, it is sure that $L \in body(r)$, for some $r \in Hyp_{i+1}$ or $L \in G_{i+1}$, so $Arg_j$ attacks $Arg_{i+1}$.

Therefore, attacks are conserved over argument rewrites. □

## 3.3 The Role of Integrity Constraints

In a cooperative dialogue, the agents attack arguments they do not agree with, but also allow themselves to be convinced otherwise by their teammates. For that reason, the better an agent explains why it disagrees with an argument, the better that agent contributes to the collective goal building a group position towards the arguments played. Therefore, in such a cooperative setting, it makes sense for agents to share the integrity constraints in their knowledge bases whenever an argument would violate it. In that case, the agent will attack with a constraint-based argument.

DEFINITION 8. *An argument* $Arg_1 = (E_1, F_1, Hyp_1, G_1)$ *is a constraint-based attack to* $Arg_2 = (E_2, F_2, Hyp_2, G_2)$ *if*

1. $C(E_1, F_1) \subseteq Lit(Arg_2)$, *minimal w.r.t. set inclusion;*
2. $G_1 = \bot$ *(to express there is an inconsistency);*
3. *There is an integrity constraint* $r_1 \in Hyp_1$ *such that* $(Hyp_1 \setminus \{r_1\}) \cup C(E_1, F_1) \models L$, *for each* $L \in body(r_1)$.

*If* $Hyp_1$ *is unitary,* $Arg_1$ *is an* impossibility attack.

Constraint-based attacks enable agents to propose the rejection of an argument or goal to the group because it does not comply with some of the agents' integrity constraints.

Let $r$ be an integrity constraint violated by $Arg$ in $\langle P, H\rangle$, i.e., all literals in the body of $r$ are true in $Arg$. To build a constraint-based attack, the agent reasons in the program $\overline{\langle P \setminus \{r\}, Lit(Arg)\rangle}$ to build an $\overline{Arg} = (\overline{E}, \overline{F}, \overline{Hyp}, \overline{G})$ that justifies the goal $\overline{G} = \bigwedge\{L \mid L \in body(r)\}$. The argument $Arg' = (\overline{E}, \overline{F}, \overline{Hyp} \cup \{r\}, \bot)$ is a constraint-based attack.

An impossibility attack is an argument that is not subject to debate and puts an end to a sequence of arguments.

EXAMPLE 7. *Consider an agent Ag with knowledge base represented by the following ALP* $\langle P, H\rangle$:

$$\begin{array}{ll} P: & a \leftarrow b; \\ & c; \\ & \leftarrow a, c. \\ H: & \{b, c\} \end{array}$$

*Consider* $A_1 = (\{\ \}, \{\ \}, \{d \leftarrow a, c;\ a;\ c\},\ d)$, *an argument that violates the integrity constraint* $\leftarrow a, c$ *in* $\langle P, H\rangle$. *The agent Ag, then, attacks* $A_1$ *with the constraint-based attack* $(\{a\}, \{\ \}, \{c; \leftarrow a, c\}, \bot)$. *Now, consider* $A_2 = (\{\ \}, \{\ \}, \{a;\ c \leftarrow a\}, G)$, $G = a, c$, *that violates the integrity constraint* $\leftarrow a, c$ *in* $\langle P, H\rangle$. *The agent cannot accept it and produces the impossibility attack* $(\{a, c\}, \{\ \}, \{\leftarrow a, c\}, \bot)$.

We highlight that constraint-based attacks considers only the facts and conclusions from the argument as abducibles.

## 3.4 Evaluating Arguments Together

For a group of agents to accept an argument, it is necessary that all attacks against it had been proven inviable, so any attacks the argument can receive should be evaluated before it. As a consequence, the first arguments accepted will be those that receive no attacks or only received conditional attacks whose conditions could not be complemented by the group. After accepting an argument, the agents should check for consequences in the acceptance of other arguments in the dialogue: An accepted argument will disqualify the ones it attacks and arguments that cannot be further attacked will get accepted.

DEFINITION 9 (ACCEPTED ARGUMENTS). *An argument* $Arg_1 = (E, F, Hyp, G)$ *is accepted by an agent with knowledge base represented by the ALP* $\langle P, H\rangle$ *if* $E = \emptyset$ *(F do not need to be empty), and*

1. *there exists an answer set S of P with which* $Arg_1$ *is consistent, i.e.,* $S \cup Lit(Arg_1)$ *does not violates any integrity constraints in P and there is no* $L \in Lit(Arg_1)$ *such that* $neg(L) \in S$*; or*

2. *every attack the agent can place against* $Arg_1$ *is defeated by another argument accepted by it.*

*A group of agents accepts an argument if all agents in the group accept it.*

Let $S$ be an answer set of $P$ and consider a $L \in S$ such that $L = neg(L')$, for some $L' \in Lit(Arg)$. The agent can build an attack $Arg' = (E, F, Hyp, L)$ in $P$ such that $E = F = \emptyset$, $Hyp \subseteq P$ and $Hyp \models L$. It is also possible that the agent can build conditional attacks against $Arg$. If the argument violates an integrity constraint in $P$, the agent should build a constraint-based attack (Section 3.3). In that case, the constraint-based attack should be played first, as it consists of an attack based in all of its answer sets.

The following theorem draws a connection between our work and *abstract argumentation* [8]. An argumentation framework is a pair $\langle S, \rho\rangle$, where $S$ is a set of arguments and $(e, f) \in \rho$, $e, f \in S$, if $e$ attacks $f$. One important concept is that of a conflict-free set of arguments, in which no argument attacks any other. An *stable extension* is a conflict-free set of arguments $S' \subseteq S$ such that every argument in $S \setminus S'$ is attacked by an element of $S'$. An argumentation framework might have zero, one or multiple stable extensions.

THEOREM 2. *If S is the set of arguments with* $E = \emptyset$ *played in a discussion, and* $\rho = \{(e, f) \in S \times S \mid e \text{ attacks } f\}$, *then the set Acc of arguments accepted by the group is a stable extension of the argumentation framework* $\langle S, \rho\rangle$.

PROOF. (Sketch) An argument can only be added to *Acc* if none of its attackers is accepted, so no other argument in *Acc* attacks it and the set is conflict-free. Furthermore, every other argument that could be accepted (with $E = \emptyset$), but is not in *Acc*, was only rejected because it is attacked by an argument in *Acc*. □

EXAMPLE 8. *Consider a group of agents engaged in a discussion on the goal* $M_0 = a, b, not\ c$. *Now suppose the arguments played are (in order):*

- $Arg_1 = (\{\ \}, \{c\}, \{a; b \leftarrow a;\}, M_0)$;
- $Arg_2 = (\{\ \}, \{\ \}, \{d; c \leftarrow d;\}, c)$ *attacks* $Arg_1$;
- $Arg_3 = (\{\ \}, \{\ \}, \{a; \neg d \leftarrow a;\}, \neg d)$ *attacks* $Arg_2$;
- $Arg_4 = (\{e\}, \{\ \}, \{e; d \leftarrow e;\}, d)$ *attacks* $Arg_3$.

*Furthermore, suppose that the agents cannot build any other arguments in the dialogue. As a result,* $Arg_4$ *is not accepted by any of the agents, since the condition e was not complemented. Because there are no other attacks against* $Arg_3$ *and it is non-conditional,* $Arg_3$ *gets accepted and disqualifies* $Arg_2$. *Since the only attack to* $Arg_1$ *was defeated, it gets accepted and so does* $M_0$. *Please note that an argument being accepted means every agent accepts it as in Definition 9. Also, note that* $\{Arg_1, Arg_3\}$ *is a stable extension of the argumentation framework involving only the arguments with* $E = \emptyset$ *and the attack relation between them.*

# 4. COOPERATIVE DIALOGUES

In this section, we investigate the acceptance of a goal or argument by a group of agents as we consider how agents deliberate individually and as a group. The satisfiability of a goal is debated as the agents place arguments to support it and others attack or rewrite them, possibly combining their knowledge in a cooperative process of group deliberation. We suppose the agents are willing to work their arguments for the best interest of the group and are honest. We also assume the agents have their knowledge bases built on the top of a common ontology and that they share the same language for communication.

The ultimate goal of the group is to build a group position towards a matter of discussion (a subject). To achieve that, the agents will take part in a dialogue, i.e., they will take turns playing arguments. The dialogue evolves through a succession of rounds in which every agent plays once, either making a move or passing. An agent will only pass if it evaluates an argument and accepts it or if it cannot play arguments. Every time a new argument is added, accepted or rejected by the group, the current matter of discussion is updated. In the first case, the recently added argument will be discussed next. If an argument is accepted or rejected, the agents will backtrack the dialogue to the last undecided matter, i.e., they will get back to further discuss the last subject that is still eligible for acceptance after receiving an attack. A single sequence of moves involving arguments in a dialogue is called a line of thought and the dialogue is the collection of lines of thought, which forms a tree with root on the initial matter of discussion.

## 4.1 Lines of Thought

DEFINITION 10. *A dialogue move is a quintuple* $Mv = (Arg, M, R, P, Agent)$ *where* $Arg$ *is an argument,* $M$ *is the matter of discussion at the time* $Mv$ *is played, and* $R \in \{att,\ sup\}$ *indicates how the move is related to* $M$*, i.e., if it attacks (att) or supports (sup)* $M$*. Similarly,* $P \in \{T, F\}$ *is the position of the argument towards the initial matter of discussion* $M_0$ *being true (*$T$*) or false (*$F$*). If* $M_0$ *is a goal,* $P = T$ *(resp.* $P = F$*) means the goal can (resp. cannot) be satisfied. If* $M_0$ *is an argument,* $P = T$ *(resp.* $P = F$*) means the argument should be accept (resp. rejected) by the group. Finally,* $Agent$ *is the author of the move.*

The initial matter of discussion is represented by a different move $Mv_0$ that might present a goal instead of an argument. Either way, this move is not based in a matter of discussion ($M = NULL$) and supports itself ($R = sup$). It also suggests the initial matter of discussion is true ($P = T$), and has no author ($Agent = NULL$). Other moves are always played by agents. These attributes of each move are kept to assure consistent reasoning during the dialogue, as well as properly backtracking the dialogues as necessary.

The following definition resembles the concepts of *argument dialogues* and *argument dialogue trees* from [1].

DEFINITION 11. *In a cooperative dialogue with k agents, a* line of thought *on a matter* $M_0$ *is a nonempty finite sequence of moves* $Mv_i = (Arg_i, M_i, R_i, P_i, Agent_i)$*,* $i \geq 0$ *such that*

1. $Mv_0 = (M_0, NULL, sup, T, NULL)$.
2. *If* $i > 0$*,* $Arg_i$ *is an argument,* $M_i$ *is a matter,* $R_i \in \{att, sup\}$*,* $P_i \in \{T, F\}$ *and* $Agent_i \in \{Ag_1, \ldots, Ag_k\}$*;*
3. *For some agent* $Ag_l$*, if* $M_0$ *is a goal, the first move played is* $Mv_1 = (Arg_1, M_0, sup, T, Ag_l)$*, since* $Arg_1$ *should justify it. Otherwise, if* $M_0$ *is an argument,* $Mv_1 = (Arg_1, M_0, att, F, Ag_l)$ *and* $Arg_1$ *attacks* $M_0$*;*
4. $Agent_{i+1} \neq Agent_i$*;*
5. *For any* $i \neq j$*,* $Arg_i$ *is a different argument from* $Arg_j$*;*
6. *If* $Arg_{i+1}$ *attacks* $Arg_i$*, then* $R_{i+1} = att$ *and* $P_{i+1} \neq P_i$*; If* $Arg_{i+1}$ *rewrites* $Arg_i$*, then* $R_{i+1} = sup$ *and* $P_{i+1} = P_i$*;*
7. *If two moves* $Mv_i, Mv_j$ *have* $P_i = P_j$*, the arguments used are consistent towards one another, i.e., there is no pair* $L$*,* $neg(L)$ *in* $Lit(Arg_i) \cup Lit(Arg_j)$*.*
8. *If* $Mv_i, Mv_j$*,* $j > i$*, are moves in the same line of thought, then* $Arg_j$ *is not attacked by* $Arg_i$*.*

*A* cooperative dialogue tree *is a finite tree with root in* $Mv_0$ *and where each branch is a line of thought. In such a tree, if two moves* $Mv_j, Mv_k$ *are played after the same* $Mv_i$ *in different lines of thought (a ramification), then* $Arg_j \neq Arg_k$*.*

A cooperative dialogue is developed as different lines of thought are explored by the agents. When an argument is played, it starts the process undecided (neither accepted nor rejected) as it might be attacked, so that argument becomes the current matter of discussion. If no agents will rewrite or attack the current matter, the line of thought reaches its end and that last argument is evaluated. The agents will then reconsider the previously played arguments in that line of thought in reverse order (backtrack), evaluating or further attacking/rewriting matters as possible. A dialogue stops when the first argument in a line of thought (other than $M_0$) is accepted. In that case, the initial matter gets satisfied (if it is a goal) or rejected (if it is an argument). Alternatively, the dialogue ends if the current matter is $M_0$, but no moves can be made to develop new lines of thought. In that case, the group can not satisfy the initial matter (if it is a goal) or has to accept it (if it is an argument).

To avoid repeating parts of the dialogue and assure consistent reasoning, the group keeps record of the sets of arguments accepted ($Acc$) and rejected ($Rej$). These sets are initially empty and are updated as arguments are evaluated by the group. We use the symbol $\Leftarrow$ to express updates.

When the group concludes the evaluation of an argument $Arg_i$ (played in the move $Mv_i$):

- If $Arg_i$ is accepted by the group, $Acc \Leftarrow Acc \cup \{Arg_i\}$. The group backtracks the dialogue to the last move $Mv_j = (Arg_j, M_j, R_j, P_j, Agent_j)$ with $P_j \neq P_i$ in the same line of thought and rejects $Arg_j$ (see below). If no such $Mv_j$ exists, the argument has the same position as the initial matter, so it is a goal that gets satisfied and the dialogue is finished.
- If $Arg_i$ is rejected by the group, $Rej \Leftarrow Rej \cup \{Arg_i\}$. The group backtracks the dialogue to the previous movement in the same line of thought and continues the dialogue. If no such movement exists, then $Arg_i$ is the initial matter, so the dialogue ends.

The dialogue tree and the sets $Acc$, $Rej$ are kept accessible to all agents (as a blackboard). An agent will only play an argument $Arg_j$ if it can still be accepted by the group, i.e., no arguments in $Acc$ attack $Arg_j$ at the time it is played.

PROPOSITION 1. *Every line of thought is finite, and so is the dialogue tree.*

PROOF. (sketch) Arguments cannot be repeated in the same line of thought. Therefore, attacks and rewrites are limited and a line of thought cannot be infinite since no cycles appear. Also, the language of the agents is finite and different lines of thought have to start with different arguments, so the dialogue tree is also finite. □

Please note that each line of thought and the sets $Acc$ and $Rej$ grow monotonically, since new arguments are introduced, but none is removed. In addition, our concept of line of thought assures the existence a stable extension (possibly more than one) over the arguments in the dialogue, as stated in Theorem 2. This is a consequence of our restrictions on what kinds of arguments can be played. Such restrictions also assure that, given a subject for discussion, the agents exhibit consistent group reasoning over two opposite positions and no argument is left undecided.

PROPOSITION 2. *The dialogue tree is developed as a depth-first search for a set of arguments accepted by the group that defines the group position towards the initial matter.*

PROOF. (sketch) The agents will always consider the last argument played to produce moves in the dialogue, and an argument is evaluated when no attacks to it can be played. That way, a single line of thought is explored at a time and possible ramifications are only considered while backtracking the arguments in a line of thought. □

## 4.2 Individual Deliberation

In a group of agents deliberating cooperatively, the parties propose arguments and collectively study the possible flaws these might have. In order to do so, agents will take turns to play arguments as they reason over two opposing positions: One that supports the initial matter (acceptance of an argument or satisfaction of a goal) and another that is against it. All agents should argue over both positions in an attempt to better explore the combination of their knowledge bases. In what follows, when we say that an agent tries to build or searches for an argument, we mean an argument that can still be played, i.e, that is not defeated by arguments previously accepted by the group.

In each turn of an agent, it will conceive available moves to play in the current line of thought. To do that, the agent considers the current matter of discussion $M$ and deliberates accordingly by attempting the following steps (in order):

- If $M$ is a goal $G$:
  1. build an argument in $P$ to justify $G$.
  2. build a conditional argument to justify $G$.
- If $M$ is an argument $Arg = (E, F, Hyp, G)$:
  1. verify if there is an $A \in Acc$ that disqualifies $Arg$;
  2. accept $Arg$, i.e., verify if it is accepted;
  3. build a constraint-based attack (Def. 8) to $Arg$;
  4. build an argument in $P$ to attack $Arg$;
  5. build a conditional argument to attack $Arg$;
  6. build an argument in $P$ to rewrite $Arg$;
  7. build a conditional argument to rewrite $Arg$;

In each case, if the agent succeeds in a step, it will play the argument built (if this is the case) and finish its turn without trying the others. If $M$ is an argument $Arg$ and the agent accepts it, the agent will pass its turn without making a move. In case an agent fails in all steps, it will also pass, for it cannot make a move.

If the current matter of discussion is a goal, the agent attempts to build an argument to justify it. If it is an argument $Arg$, but it is not consistent with an answer set of $P$, the agent will try to attack it. In that case, the agent will first attempt constraint-based attacks, since violating an integrity constraint means the argument might be inconsistent with multiple answer sets. Next, the agent tries to build an attack (non-conditional) in $P$, based on an answer set. Finally, the agent appeals to explanations and conditional arguments to disqualify the argument in question. If the argument is conditional with $E \neq \emptyset$, the agent should try to rewrite the argument. If no attacks can be played and the argument has $E = \emptyset$, the agent has to accept it (Definition 9), so it passes its turn (no moves available).

## 4.3 Dialogue Example

Our dialogue framework proposes a model for group deliberation that is fair as all agents have the same number of chances to play arguments in the discussion. In this process, agents cannot only state their individual arguments and demands on each matter of discussion, but also combine their knowledge to build collective arguments. These agents take turns placing arguments and might get convinced by their colleagues to accept opinions they would not if they were on their own. Next, we show an example of dialogue with two lines of thought. For an easier comprehension of the example, we will only show the arguments involved in each move. We will show the updates of $Acc$, $Rej$ and the current matter of discussion.

EXAMPLE 9. *Consider a group of three agents engage in a discussion on the matter $M_0 = a, b, not\ c$. In the sequel, we will list the arguments placed by the agents and write (round, turn) to enumerate them.*

*(1, 1) $Ag_1$ plays $Arg_1 = (\{\}, \{c\}, \{a; b \leftarrow a;\}, M_0)$;*

*(1, 2) $Ag_2$ attacks $Arg_1$ with $Arg_2 = (\{\}, \{\}, \{d; c \leftarrow d;\}, c)$;*

*(1, 3) $Ag_3$ attacks $Arg_2$ with $Arg_3 = (\{\}, \{\}, \{a; \neg d \leftarrow a;\}, \neg d)$;*

*(2, 1) $Ag_1$ accepts $Arg_3$ and passes;*

*(2, 2) $Ag_2$ attacks $Arg_3$ with $Arg_4 = (\{e\}, \{\}, \{e; d \leftarrow e;\}, d)$;*

*(2, 3) $Ag_3$ cannot attack or rewrite $Arg_4$ (pass);*

*(3, 1) $Ag_1$ cannot attack or rewrite $Arg_4$ (pass). $Rej \Leftarrow Rej \cup \{Arg_4\}$, $M \Leftarrow Arg_3$;*

*(3, 2) $Ag_2$ attacks $Arg_3$ with $Arg_5 = (\{\}, \{\}, \{\neg a \leftarrow not\ a;\}, \neg a)$;*

*(3, 3) $Ag_3$ attacks $Arg_5$ with $Arg_6 = (\{\}, \{\}, \{a;\}, a)$;*

*(4, 1) $Ag_1$ accepts $Arg_6$ and passes;*

*(4, 2) $Ag_2$ cannot attack or rewrite $Arg_6$ (pass). $Acc \Leftarrow Acc \cup \{Arg_6\}$, $Rej \Leftarrow Rej \cup \{Arg_5\}$, $M \Leftarrow Arg_3$;*

*(4, 3) $Ag_3$ accepts its own argument $Arg_3$ (pass);*

*(5, 1) $Ag_1$ accepts $Arg_3$ and passes;*

*(5, 2) $Ag_2$ cannot attack or rewrite $Arg_3$ (pass). $Acc \Leftarrow Acc \cup \{Arg_3\}$, $Rej \Leftarrow Rej \cup \{Arg_2\}$, $M \Leftarrow Arg_1$;*

*(5, 3) $Ag_3$ accepts $Arg_1$ (pass);*

*(6, 1) $Ag_1$ accepts $Arg_1$ (pass);*

*(6, 2) $Ag_2$ accepts $Arg_1$, for it cannot attack or rewrite it, and passes its turn. $Acc \Leftarrow Acc \cup \{Arg_1\}$.*

*As a result of the acceptance of $Arg_1$, the initial matter $M_0$ is also accepted and the discussion is finished.*

In each step of the dialogue, the current matter is updated to the last argument placed (after a move) or left undecided (after arguments get evaluated). An argument is evaluated if a full round passes and the agents do not make any moves. In Example 9, if $Ag_2$ were able to attack $Arg_3$ on step (5,2) or $Arg_1$ on step (6,2), the dialogue would continue on a different line of thought. Please note that $Ag_2$ cannot use $Arg_5$ to attack the arguments $Arg_3$ and $Arg_1$ in different lines of thought because it has been rejected in step (4,2).

# 5. RELATED WORK

Argumentative Deliberation [13] involves the use of arguments by agents to support self deliberation and also employs abductive reasoning. This approach introduces conditional arguments that are played in a dialogue, but the abductive hypotheses are not shared as such. In a cooperative setting, however, it makes sense to share them with other agents. In our work, a group of agents can share hypothesis to combine their knowledge and produce interesting arguments that they would possibly not be able to conceive individually. Judgement Aggregation [4] allows agents in a group to combine their individual judgements over a set of arguments and collectively decide which ones to accept. Unlike our work, this approach does not consider communication amongst the agents. Abductive reasoning and argumentation have also been combined together in [2, 15]. In both papers, the explanations and arguments are produced by a single agent at a time and their knowledge is not combined. In our work, agents share their hypothesis to combine their knowledge and reach consensus over acceptable arguments and a group position towards a matter of discussion. In [9, 5, 3] agents can share hypotheses to produce group explanations. In our proposal, agents will provide arguments to support or attack each others hypotheses. Our goal, however, is not to produce group explanations with combined hypotheses, but for agents to point out missing pieces of their arguments, which can be complemented or criticized by others. The works in [6, 18] study collaborations in distributed argumentation as agents form coalitions to produce group arguments. They consider partial arguments, which are partial derivations of arguments that need complimentary knowledge. Althought unclaimed, the kind of reasoning they introduce is clearly abductive. Our work innovates as group arguments are built in a dialogue and extensively discussed by the agents, so the conditional arguments, which are much similar to partial arguments, might also be attacked and are subject to rejection. Another important difference is that our agents are able to detect possible inconsistencies amongst their beliefs as we recur to extended abduction [16, 17].

## 6. CONCLUSION AND FUTURE WORKS

We presented an approach to cooperative dialogues in groups of agents. In our framework, agents can play arguments and attack the opinions of each other, but can also complement them and build more elaborate ones. This innovative feature allows agents to rewrite arguments and combine their knowledge as they search for an unified group opinion about some matter of discussion and their own arguments. We have enabled such a cooperative behavior by employing abductive reasoning and changing the way that abduction-based conditional arguments are placed in a dialogue. This cooperative behavior can be also perceived as group deliberation. Even though we only consider arguments and goals as initial matters of discussion, a group can deliberate about how to accomplish agent goals, evaluate proposals in a negotiation, make group decisions, and so on, giving our framework a number of different applications. In our future works, we will explore these applications, their particularities, and study what kinds of roles the individual preferences of agents should play in the process.

## 7. ACKNOWLEDGEMENTS

Research partially supported by CAPES (PROCAD).

## 8. REFERENCES

[1] L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In W. Horn, editor, *ECAI*, pages 338–342. IOS Press, 2000.

[2] F. Bex and H. Prakken. Investigating stories in a formal dialogue game. In *Proceeding of the Conference on Computational Models of Argument (COMMA 2008)*, pages 73–84. IOS Press, 2008.

[3] G. Bourgne, K. Inoue, and N. Maudet. Abduction of distributed theories through local interactions. In H. Coelho, R. Studer, and M. Wooldridge, editors, *ECAI*, volume 215 of *Frontiers in Artif. Intell*, pages 901–906. IOS Press, 2010.

[4] M. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[5] A. Ciampolini, E. Lamma, P. Mello, F. Toni, and P. Torroni. Cooperation and competition in alias: A logic framework for agents that negotiate. *Ann. Math. Artif. Intell.*, 37(1-2):65–91, 2003.

[6] I. de Almeida Móra, J. J. Alferes, and M. Schroeder. Argumentation and cooperation for distributed extended logic programs. In *Nonmonotonic Reasoning Workshop*, 1998.

[7] M. Denecker and A. C. Kakas. Abduction in logic programming. In A. C. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond*, volume 2407 of *LNCS*, pages 402–436. Springer, 2002.

[8] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[9] M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. An abductive framework for information exchange in multi-agent systems. In J. Dix and J. A. Leite, editors, *CLIMA*, LNCS, vol. 3259, pages 34–52. Springer, 2004.

[10] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Comput.*, 9(3/4):365–386, 1991.

[11] K. Inoue and C. Sakama. Abductive framework for nonmonotonic theory change. In *Proc. 14th International Joint Conference on Artificial Intelligence - Volume 1*, pages 204–210, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[12] A. C. Kakas, R. A. Kowalski, and F. Toni. The role of abduction in logic programming. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook in Artificial Intelligence and Logic Programming*, volume 5 of *Datalogiske Skrifter*, pages 235–324. Roskilde University, 1998.

[13] A. C. Kakas and P. Moraitis. Argumentative agent deliberation, roles and context. In J. Dix, J. A. Leite, and K. Satoh, editors, *CLIMA*, Datalogiske Skrifter, vol. 93, pages 35–48. Roskilde University, 2002.

[14] P. McBurney, D. Hitchcock, and S. Parsons. The eightfold way of deliberation dialogue. *Int. J. Intell. Syst.*, 22(1):95–132, 2007.

[15] F. Sadri, F. Toni, and P. Torroni. Logic agents, dialogues and negotiation: An abductive approach. In *In Proc. AISB'01 Convention.* AISB, 2001.

[16] C. Sakama and K. Inoue. An abductive framework for computing knowledge base updates. *Theory Pract. Log. Program.*, 3:671–715, November 2003.

[17] C. Sakama and K. Inoue. Negotiation by abduction and relaxation. In *Proc. 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS'07, pages 242:1–242:8, New York, NY, USA, 2007. ACM.

[18] M. Thimm, A. J. Garcia, G. Kern-Isberner, and G. R. Simari. Using collaborations for distributed argumentation with defeasible logic programming. In M. Pagnucco and M. Thielscher, editors, *NMR'08*, pages 179–188. University of New South Wales, Technical Report No. UNSW-CSE-TR-0819, September 2008.