Formal properties of the $\mathcal{S}$CIFF-AF Multi-Agent Argumentation Framework*

Paolo Torroni

DEIS, University of Bologna
V.le Risorgimento 2, 40136 Bologna, Italy
Paolo.Torroni@UniBO.it

Abstract. Argumentation theories have recently emerged and gained popularity in the agents community, since argumentation represents a natural and intuitive way to model non-monotonic reasoning. In a multi-agent context, argumentation has recently been proposed as a component of dialogue frameworks. However, despite the large interest in argumentation theories in multiagent domains, most proposed frameworks stay at a general though abstract level, and operational counterparts to abstract frameworks are not many. The aim of this work is to present the main formal properties of the $\mathcal{S}$CIFF-AF: an operational argumentation-based multiagent dialogue framework.

1 Introduction

Argumentation theories have recently emerged and gained popularity in the agents community, since argumentation represents a natural and intuitive way to model non-monotonic reasoning. In a multiagent context, argumentation has recently been proposed as a component of dialogue frameworks. A typical setting is that of collaborative problem solving, for example to tackle resource allocation and achievement [18]. In such a context, multiple agents have to coordinate in order to take joint decisions about possible allocations of resources.

In general, argumentative reasoning can be utilized by agents intending to decide about possible future courses of action. Typically, in collaborative problem solving domains, individual agents have their own goals to achieve and their own constraints to satisfy, but they are situated in a common environment in which there are resources they need to share. Thus when agents take actions they need to ensure that their activity does not clash with other agents' actions and constraints.

We then have to consider two aspects of collaborative problem solving: from an individual's perspective, an agent should be able to reason about what is the most appropriate course of actions to take in a given situation. We believe that the theories and logics of argumentation are a very promising approach to this problem. From a "social" perspective, instead, agents can use argumentation in

* This is an improved version of an article discussed at the LPNMR Workshop on Argumentation and NonMonotonic Reasoning, held in Tempe, AZ, on May 14, 2007.

order to engage in dialogues, and use their arguments to make their decisions accepted by other agents.

The first aspect is related to deliberation and practical reasoning, central issues in agent architectures and reasoning since the early days of BDI agent models and implementations [17, 9] and further pursued in recent work such as [15, 11, 5]. The intuition behind the use of argumentation theories for deliberation is that when several alternative courses of actions are possible, and an agent must decide which way to go, in order to evaluate its options the agent will try to find supporting arguments for the one or the other option, and decide based on the presence of supporting arguments and, like in [11], possibly based on preferences among arguments. In general, arguments are not used in isolation, but in conjunction with other arguments, to form sets of arguments. Such sets are said to be "admissible" if they prevail over all their possible counterarguments. In this way, the deliberation problem is mapped onto an admissibility check problem, enriched sometimes with preference reasoning.

The aspect of argumentation in agent dialogue has also been addressed by conspicuous work of literature [4, 18, 14], and considered by many as the natural evolution of agent dialogue in domains such as negotiation [16]. In multiagent research, the idea behind the use of argumentation in this context is that some types of dialogues, like for instance persuasion and negotiation dialogues, can be considered as dialogue games in which players construct arguments to support their claims and convince their opponent about the truth of a proposition or about the convenience of an action. We can consider this kind of joint activity as a special case of collaborative problem solving. Also in this research area, the literature is vaste. Among others, and closely related to the research presented here, Atkinson *et al.* explore the issue of multiagent argument over proposals for action [5].

Despite the large interest in argumentation theories in multiagent domains, most frameworks of literature are proposed at a general though abstract level, and without an operational counterpart. One important contribution which goes in the direction of reconciling abstract argumentation with proof-theoretic frameworks is work by Kakas and Toni [12] on mapping Dung's abstract argumentation framework [7] onto the Abductive Logic Programming (ALP) framework [10].

Motivated by [12] and drawing inspiration from Atkinsons *et al.*'s work about the PARMA action persuasion protocol [5], we have proposed an Argumentation Framework [20] based on the ALP $\mathcal{S}$CIFF framework ($\mathcal{S}$CIFF-AF) for multiagent argumentation, aimed at narrowing down the gap between abstract and operational argumentation frameworks in the multiagent domain. $\mathcal{S}$CIFF-AF encompasses multiagent dialogues over proposals for action, and it is equipped with a declarative and operational model with an ALP semantics.

The formal foundations of this framework rely on previous results from ALP, and from Dung's studies on argumentation. Basically, $\mathcal{S}$CIFF-AF is a casting of Dung's abstract argumentation framework in $\mathcal{S}$CIFF, in the style of [12], augmented with a notion of 2-party agent dialogue and agreement over proposals for actions. In fact, agent dialogues in $\mathcal{S}$CIFF-AF can be used by the interacting

parties to reach a consensus on a possible future course of action and consequent state, and ultimately such actions may be adopted by agents as future internal goals.

An example that shows how $\mathcal{S}$CIFF-AF could be used is the following. Consider two agents, s and d, the first one representing a scientist, and the second one his department. s holds a knowledge base, which represents his program. s's goal is: to attend a conference. In order to decide a possible course of actions (plan) leading to the achievement of his goal, s reasons in an argumentative fashion, and considers arguments in favour/against this or that plan. Among all the possible sets of arguments, s will have to decide whih ones make a "good" justification for its goal and plan, i.e., which ones are *admissible*. A possible plan could involve flying to the conference venue in business class. s consider this to be a suitable plan, backed-up by admissible arguments. Unfortunately, d is not of the same opinion, since d has arguments against s flying business class. s's goal can be reconciled with d's position via a dialogue, in which s and d, collaboratively, reach an agreement about a possible plan of actions achieveing s's goal, which can be agreed upon by both agents.

As we can see from this example, we need a language to express an agent's knowledge and to model actions, a formalism to represent arguments, a framework to reason upon them and a semantics of admissibility. We need a way to compute admissible sets of arguments, to represent exchange of arguments, and to identify agreements about possible courses of actions. A framework that exhibits such features is $\mathcal{S}$CIFF-AF. The aim of this work is to present its main formal properties, which are important to insure a consistent and meaningful system evolution. We start by showing semantic properties of the argumentation framework in relation with Dung's argumentation semantics. Later, we refine the definition of multiagent dialogue proposed in [20] and we show what properties multiagent agreements exhibit. We conclude by showing the implementation of the example above.

2 Background

$\mathcal{S}$CIFF-AF is built on three pillars: Dung's abstract argumentation framework [7], the $\mathcal{S}$CIFF language and Abductive Logic Programming (ALP) proof-procedure [2], and the PARMA action persuasion protocol and its locutions [5].

ALP is a computational paradigm aimed to introduce hypothetical reasoning in the context of Logic Programming (see [13] for an introduction to LP and [10] for a survey on ALP). A logic program $\mathcal{P}$ is a collection of clauses, with an associated notion of entailment, usually indicated by $\models$. In ALP, some predicates ("*abducibles*"), belonging to a special set $\mathcal{A}$, can be assumed to be true, if need be. In order to prevent unconstrained hypothesis-making, $\mathcal{P}$ is typically augmented with expressions which *must be true at all times*, called *integrity constraints* ($\mathcal{IC}$). An *abductive logic program* is the triplet $\langle \mathcal{P}, \mathcal{A}, \mathcal{IC} \rangle$, with an associated notion of abductive entailment.

$\mathcal{S}$CIFF is an ALP proof-procedure defined by Alberti *et al.* [2] as an extension of Fung and Kowalski's IFF [8], and it is the reference ALP framework for this work. One distinguishing feature of $\mathcal{S}$CIFF is its notion of *expectations* about events. Expectations are abducibles denoted as $\mathbf{E}(X)$ (*positive* expectations) and $\mathbf{EN}(X)$ (*negative* expectations), where $\mathbf{E}(X)/\mathbf{EN}(X)$ stand for "X is expected/expected not to happen". Variables in events, expectations and in other atoms can be subject to CLP constraints and quantifier restrictions.

Two fundamental concepts in $\mathcal{S}$CIFF are those of *consistency* and *entailment*. We report their definition below, taken from [2].

Definition 1 (Consistent sets of hypotheses). *A set of hypotheses Δ is* consistent *if and only if* $\forall$ (ground) $p, \{p, not\, p\} \nsubseteq \Delta$ *and* $\{\mathbf{E}(p), \mathbf{EN}(p)\} \nsubseteq \Delta$.

Definition 2 (Entailment). *A ($\mathcal{S}$CIFF) ALP $\mathcal{S} = \langle \mathcal{P}, \mathcal{A}, \mathcal{IC} \rangle$* entails *a goal G (written $\mathcal{S} \models_\Delta G$), if and only if:*

$$\begin{cases} Comp(\mathcal{P} \cup \Delta) \cup CET \cup T_\chi \models G\sigma \\ Comp(\mathcal{P} \cup \Delta) \cup CET \cup T_\chi \models \mathcal{IC} \end{cases}$$

where Comp is the symbol of completion, CET is Clark's equality theory, $\models$ is Kunen's logical consequence relation for three-valued logic, σ is a substitution of ground terms for the variables in G, T_χ the theory of constraints, and Δ a consistent *subset of $\mathcal{A}$.*

$\mathcal{S}$CIFF operates by considering G together with $\mathcal{IC}$ as the initial goal, and by calculating a *frontier* as a disjunction of conjunctions of formulae, using at each step one among the inference rules defined in [2]. Given the frontier, at any step a selection function can be used to pick one among all the equally true disjuncts in the frontier. When no more inference rule applies (*termination*), if there exists at least one disjunct which is not false, then $\mathcal{S}$CIFF has succeeded, and Δ contains an answer to G. The $\mathcal{S}$CIFF proof-procedure is sound, and under reasonable restrictions it is also complete [2]. $\mathcal{S}$CIFF has been implemented and instantiated into a large number of scenarios involving agent communication, and it can be downloaded from its web site.[1]

From now on, if not explicitly mentioned otherwise, we will always refer to an arbitrary but fixed instance $\mathcal{S} = \langle \mathcal{P}, \mathcal{A}, \mathcal{IC} \rangle$ of a $\mathcal{S}$CIFF abductive framework.

Following Kakas and Toni [12], in $\mathcal{S}$CIFF-AF arguments are mapped onto abducibles. For example, an assumption $\mathbf{E}(p)$, "p is expected", could be considered as an argument which possibly supports some goal g.

Definition 3 (Argument). *An* argument *is a literal p or not p, where p is an abducible atom.*

Arguments could be therefore any (possibly negated) element of $\mathcal{A}$, including expectations in the form $\mathbf{E}(t)/\mathbf{EN}(t)$, where t is a term. By arguments we can model circumstances (in the sense of [5]), actions, and related constraints.

[1] `http://lia.deis.unibo.it/research/sciff/`.

Thus an agent may justify a goal g by saying, e.g., "in order to achieve a goal g, under the circumstances c and the constraints x, actions a_1 and a_2 should be carried out." In order to take this kind of position, an agent will utter the various elements of it (the circumstances, the goal, the actions, the constraints) via a suitable argumentation language and using the appropriate locutions. Argumentation dialogues will provide implicit links among such uttered elements.

Our proposed argumentation framework is an instantiation of Dung's work [7] and of the abstract computational framework developed by Kakas and Toni [12]. In particular, Dung's notion of *attack* is rephrased in the following way:

Definition 4 (Attacks). *A set of arguments A attacks another set Δ if and only if at least one of the following expressions is true:*

(1) $\mathcal{S} \models_A not\, p$*, for some* $p \in \Delta$*;*
(2) $\mathcal{S} \models_A \mathbf{E}(p)$*, for some* $\mathbf{EN}(p) \in \Delta$*;*
(3) $\mathcal{S} \models_A \mathbf{EN}(p)$*, for some* $\mathbf{E}(p) \in \Delta$*;*

Definition 5 (Argumentation Framework). *An Argumentation Framework (AF) is the pair* $\langle \mathcal{S}, attacks \rangle$*.*

In a multiagent context, agents can locally reason about circumstances, constraints, and actions (not) to be taken, based on the $\mathcal{S}$CIFF-AF, and produce – at the social level – dialogues in the style of PARMA dialogues.

PARMA considers a general argument schema for a rational position proposing an action, and handles possible attacks on one or more elements of a general argument schema. Attacks arise from disagreements originating from different sources. PARMA uses four categories of *locutions*, for dialogue control (C), action proposal (P), inquiry (A), and denial (D) of existence/validity of elements of a position. Such elements could be goals, circumstances, and actions (not) to be taken. While Atkinson *et al.* focus on addressing divergences on all elements of a position, $\mathcal{S}$CIFF-AF focusses instead on a more restricted number of issues, and adopts only a small set of locutions. In particular, it only considers some control locutions (C) and some proposal/denial locutions about circumstances and actions (P/D).

Definition 6 (Agent system). *An* agent system *is a finite set Σ, where each $x \in \Sigma$ is a ground term, representing the name of an agent, equipped with a $\mathcal{S}$CIFF program $\mathcal{S} = \{\mathcal{P}, \mathcal{A}, \mathcal{IC}\}$.*

Definition 7 (Performative or dialogue move). *A* performative *or* dialogue move *p is an instance of a schema tell$(a, b, L[, Arg])$, where a is the* utterer*, b is the* receiver*, L is the* locution *and (optionally) Arg is the* argument *of the performative. For a given p, utterer$(p) = a$, receiver$(p) = b$, locution$(p) = L$ and argument$(p) = Arg$ (if present). The set of all possible performatives is called* argumentation language.

Note that Arg is optional, since a dialogue move not necessarily contains arguments all the time. In general, dialogue control (C) locutions will not need it.

For instance, at start, an agent may simply want to declare that he is listening. In the definition below, we gear $\mathcal{S}$CIFF-AF with a concrete argumentation language inspired by PARMA.

Definition 8 (The argumentation language $\mathcal{L}_{arg}$). *The* argumentation language $\mathcal{L}_{arg}$ *is the set of all performatives p, such that:*

- *locution*$(p) \in \{$ *'enter dialogue'*, *'leave dialogue'*, *'term finished'*, *'accept denial'*, *'state circumstances'*, *'deny circumstances'*, *'state actions'*, *'deny actions'* $\}$, *and*
- *argument*(p) *is a conjunction of abducible atoms (possibly including* **E/EN** *expectations) and CLP constraints.*

$\mathcal{S}$CIFF-AF thus defines a concrete language for argumentation, $\mathcal{L}_{arg}$, which includes four dialogue control locutions (type C: *'enter dialogue'*, *'leave dialogue'*, *'term finished'*, and *'accept denial'*), two proposal locutions (P: *'state circumstances'* and *'state actions'*) and two denial locutions (D: *'deny circumstances'* and *'deny actions'*). Agents conversing in $\mathcal{L}_{arg}$ will not exchange formulae stating e.g. consequences of actions, such as implications, but only conjunctions of atoms.

Definition 9 (MAS argumentation framework). *A* MAS argumentation framework $\mathcal{M}$ *is a pair* $\langle \Sigma, Actions \rangle$ *where* Σ *is a multiagent system of agents with the same* $\mathcal{A}$ *which communicate using* $\mathcal{L}_{arg}$*, and Actions is a finite set, where each element is a ground term, representing the name of an action.*

Beside assuming a common language, $\mathcal{S}$CIFF-AF also assumes a common ontology (thus in Definition 9 $\mathcal{A}$ is the same for all agents in Σ). Otherwise some ontological middleware may be used so that, for example, in a position involving a sales, "buy" and "purchase" converge down to the same meaning. This is most necessary in open systems, to prevent misunderstandings arising from the use of terminology. Note that the presence of an argumentation framework based on ALP does not prevent agents from having and reasoning upon their private knowledge, and especially it does not prevent them from having private abducibles. However, for the sake of simplicity, in this article we will focus only on those abducibles which are functional to agent dialogue, and we assume that such abducibles are common to all agents for the reasons above.

In [20] argumentation dialogues are defined between two agents, and their evolution is modelled as a sequence of states. Each state contains a set of arguments modelling stated/agreed circumstances and actions, and possibly agreements reached by the agents.

3 Properties of SCIFF-AF

In this section we refine the original $\mathcal{S}$CIFF-AF framework. The idea is to define a notion of agent agreement about actions, and focus on the fundamental properties of multiagent agreements in $\mathcal{S}$CIFF-AF. Before doing so, we also discuss some important semantic properties of the $\mathcal{S}$CIFF-AF framework.

3.1 Admissible sets and grounded semantics of SCIFF-AF

Let us consider the *attacks* relation taken from [20] and reported in Section 2.

Lemma 1 *The following propositions are true:*

- *No set of arguments attacks the empty set of arguments* $\emptyset$*;*
- *attacks is* monotonic*, i.e. for all (consistent)* A*,* A'*,* Δ*,* $\Delta' \subseteq \mathcal{A}$*, if* A *attacks* Δ *then*
  - (i) *if* $A \subseteq A'$ *then* A' *attacks* Δ*, and*
  - (ii) *if* $\Delta \subseteq \Delta'$ *then* A *attacks* Δ'*;*
- *attacks is* compact*, i.e. for all* A*,* $\Delta \subseteq \mathcal{A}$*, if* A *attacks* Δ *then there exists a finite* $A' \subseteq A$ *such that* A' *attacks* Δ*;*

Proof. The first proposition follows from the definition of *attacks*. The second proposition follows from the fact that if $\mathcal{S} \models_A not\, p$, for some $p \in \Delta$, then $\forall A' \supseteq A, p \in A'$, therefore A' attacks Δ (i), and $\forall \Delta' \supseteq \Delta, not\, p \in \Delta'$, therefore A' attacks Δ (ii). The same holds if the attack is on some $\mathbf{E}(p)/\mathbf{EN}(p)$. The third proposition follows from the compactness of $\models_A$, by which finite expressions are always derivable from a finite set of antecedents.

These properties are considered by Kakas and Toni fundamental of an attacking relation [12, pag.518].

Remark 1. The notion of *attacks* introduced is symmetric, which makes of $\mathcal{S}$CIFF-AF a symmetric argumentation framework. Moreover, *attacks* is irreflexive and in all non trivial cases non empty. This has some positive consequences, which lead to the agreement of several semantics, and to the fact that $\mathcal{S}$CIFF-AF is a coherent and grounded framework [6]. However, we will focus on the admissible sets semantics, since it suffices for our purposes.

Remark 2. For an argument A such that $\mathcal{S} \models_A p$ for some p, it follows from the declarative semantics of $\mathcal{S}$CIFF that A is consistent, and that if an argument Δ is attacked by A, $A \cup \Delta$ is not consistent (in the sense of $\mathcal{S}$CIFF).

The definitions that follow are taken from Dung's abstract argumentation framework [7]. Corollaries 1 and 2 show the results of its instantiation in the $\mathcal{S}$CIFF framework.

Definition 10 (Conflict-free sets of arguments). *A set* Δ *of arguments is said to be* conflict-free *if there are no arguments* A *and* B *in* Δ *such that* A *attacks* B*.*

Corollary 1. *All consistent sets of arguments (in the sense of* $\mathcal{S}$*CIFF) are conflict-free.*

Proof. Let A be one among $\{not\, p, \mathbf{E}(p), \mathbf{EN}(p)\}$, and let $\hat{A}$ be the corresponding "attacked" argument ($p, \mathbf{EN}(p)$, or $\mathbf{E}(p)$, respectively). Let Δ be a consistent set of arguments, and A, B two arguments in Δ. A *attacks* B means $\mathcal{S} \models_{\Delta'} A$ for some Δ', and $B = \hat{A}$; but this would imply that $\{A, \hat{A}\} \subseteq \Delta$. Contradiction!

As a consequence of of Remark 2 and Corollary 1, we have:

Corollary 2. *All arguments A such that $\mathcal{S} \vDash_A p$ for some p are conflict-free.*

Finally, admissible sets of arguments are defined following Dung [7, Definition 6] and Kakas & Toni [12, Definition 2.3].

Definition 11 (Admissible set of arguments). *A (conflict-free) set of arguments Δ is* admissible *iff for all sets of arguments A, if A attacks Δ, then Δ attacks $A \setminus \Delta$.*

Dung's Fundamental Lemma [7, pag. 327], together with the fact that the empty set is always admissible, implies the following corollary:

Corollary 3. *All arguments A such that $\mathcal{S} \vDash_A p$ for some p are admissible sets of arguments for $\mathcal{S}$.*

Dung defines *preferred extensions* as maximal sets of admissible sets of arguments [7, Definition 7], but we will focus on admissible sets of arguments rather than on preferred extensions. In fact, as stressed by Kakas and Toni [12], since every admissible set of arguments is contained in some preferred extension, in order to determine whether a given query holds with respect to the preferred extension and partial stable model semantics, it is sufficient to determine whether the query holds with respect to the semantics of admissible sets.

Finally, the IFF proof-procedure upon which $\mathcal{S}$CIFF is built has a grounded argumentation semantics. Therefore we can conclude this section with a last important semantic property of the $\mathcal{S}$CIFF-AF framework.

Corollary 4. *All arguments A such that $\mathcal{S} \vDash_A p$ for some p are grounded sets of arguments for $\mathcal{S}$.*

3.2 Properties of SCIFF-AF dialogues and agreements

In this section, we specialize the $\mathcal{S}$CIFF-AF dialogue framework, to define precisely what multiagent agreements are, and to show what properties they exhibit. The following definitions are based on the notions of agent system, performative, argumentation language and MAS argumentation framework given in Section 2.

Definition 12 (Dialogue). *Given an agent system Σ, a* dialogue $\mathcal{D}$ *in a language $\mathcal{L}$, between two agents $x, y \in \Sigma$, is an ordered set of performatives $\{p_0, p_1, \ldots\} \subseteq \mathcal{L}$, such that $\forall p_j = tell(a_j, b_j, L_j, A_j) \in \mathcal{D}, (a_j, b_j) \in \{(x, y), (y, x)\}$*

An example of dialogue will be provided later on (Example 1). The one above is a general definition, and it can be instantiated by choosing a concrete language, e.g. $\mathcal{L} = \mathcal{L}_{arg}$.

Definition 13 (State of a dialogue in $\mathcal{L}_{arg}$). *Given a dialogue $\mathcal{D}$ in $\mathcal{L}_{arg}$, for each j, $1 < j < |\mathcal{D}|$ the* state of the dialogue*, state($\mathcal{D}, j$) is a tuple*

$$\langle \Psi_j^{sc}, \Psi_j^{dc}, \Psi_j^{sa}, \Psi_j^{da}, \Psi_j^{aa} \rangle,$$

defined based on the dialogue history $\mathcal{D}_j = \{p_0, p_1, \ldots, p_{j-1}\}$ *as follows:*

– Ψ_j^{sc} *is the* set of stated circumstances*, defined as:*
$\Psi_j^{sc} = \{\ circ$ such that $\exists p_k \in \mathcal{D}_j\ \wedge\ k < j$
$\wedge$ *locution*$(p_k) =$ *'state circumstances'* $\wedge$ $circ \in$ *argument*(p_k)
$\wedge\ \nexists p_l \in \mathcal{D}_j\ \wedge\ k < l < j$ such that (
locution$(p_l) =$ *'state circumstances'* $\wedge$ *argument*$(p_k) \neq$ *argument*(p_l)) $\}$
– Ψ_j^{dc} *is the* set of denied circumstances*, defined as:*
$\Psi_j^{dc} = \{\ circ$ such that $\exists p_k \in \mathcal{D}_j\ \wedge\ k < j$
$\wedge$ *locution*$(p_k) =$ *'deny circumstances'* $\wedge$ $circ \in$ *argument*(p_k)
$\wedge\ \nexists p_l \in \mathcal{D}_j\ \wedge\ k < l < j$ such that *locution*$(p_l) =$ *'state circumstances'* $\}$
– Ψ_j^{sa} *is the* set of stated actions*, defined as:*
$\Psi_j^{sa} = \{\ \mathbf{E}(act)$ such that $\exists p_k \in \mathcal{D}_j\ \wedge\ k < j$
$\wedge$ *locution*$(p_k) =$ *'state actions'* $\wedge$ $\mathbf{E}(act) \in$ *argument*(p_k)
$\wedge\ \nexists p_l \in \mathcal{D}_j\ \wedge\ k < l < j$ such that (
locution$(p_l) =$ *'state actions'* $\wedge$ *argument*$(p_k) \neq$ *argument*(p_l)) $\}$
– Ψ_j^{da} *is the* set of denied actions*, defined as:*
$\Psi_j^{da} = \{\ \mathbf{E}(act)$ such that $\exists p_k \in \mathcal{D}_j\ \wedge\ k < j$
$\wedge$ *locution*$(p_k) =$ *'deny actions'* $\wedge$ $\mathbf{E}(act) \in$ *argument*(p_k)
$\wedge\ \nexists p_l \in \mathcal{D}_j\ \wedge\ k < l < j$ such that *locution*$(p_l) =$ *'state actions'* $\}$
– Ψ_j^{aa} *is the* set of agreed actions*, defined as:*
$\Psi_j^{aa} = \{\ \mathbf{E}(act)$ such that $\exists p_k, p_l \in \mathcal{D}_j$
$\wedge\ k < j\ \wedge\ l < j\ \wedge$ *locution*$(p_k) =$ *locution*$(p_l) =$ *'state actions'*
$\wedge$ *argument*$(p_k) =$ *argument*$(p_l)\ \wedge\ \mathbf{E}(act) \in$ *argument*$(p_k)\ \}$

By Definition 13, the state of the dialogue at a step j with respect to circumstances/actions is determined by the last relevant move made.

Note that *state*(D, j) is defined independently of control locutions, and that locutions *'state circumstances'* and *'state actions'* operate some sort of reset of the current state: if an agent utters *'state circumstances'* at step j, the set of stated circumstances will only contain the new circumstances Ψ_j^{sc}, until some agent again states *'state circumstances'*, and *'deny circumstances'* becomes the empty set, since the previously denied circumstances become obsolete. A similar semantics is that of *'state actions'* and *'deny actions'*. Note that memory of past moves is not necessarily lost, since agents may reason based on the previous states.

This definition of state is a specialization of the one given in [20]. We can immediately see what structural properties it exhibits:

Corollary 5. *Given a dialogue $\mathcal{D}$ in $\mathcal{L}_{arg}$, the state of $\mathcal{D}$ at step j,* state$(\mathcal{D}, j) = \langle \Psi_j^{sc}, \Psi_j^{dc}, \Psi_j^{sa}, \Psi_j^{da}, \Psi_j^{aa} \rangle$*, enjoys the following structural properties:*

1. $\Psi_j^{dc} \subseteq \Psi_j^{sc}$ *("coherence" between the set of denied circumstances and the set of stated circumstances)*
2. $\Psi_j^{da} \subseteq \Psi_j^{sa}$ *("coherence" between the set of denied actions and the set of stated actions)*
3. $\Psi_j^{aa} = \Psi_j^{sa} \vee \Psi_j^{aa} = \emptyset$ *("coherence" between the set of agreed actions and the set of stated actions)*

Proof. The proof follows from Definition 13.

We can now proceed with defining the central concept of argumentation dialogue, which is as well a specialization of the one proposed in [20].

Definition 14 (Argumentation Dialogue). *Given a multiagent argumentation framework* $\mathcal{M} = \langle \Sigma, Actions \rangle$*, an* argumentation dialogue $\mathcal{D}$ *between* $x, y \in \Sigma$*, respectively equipped with* $\mathcal{S}^x/\mathcal{S}^y$*,* about a goal G_x *is a dialogue in* $\mathcal{L}_{arg}$ *such that:*

1. $p_0 = tell(x, y, \text{'enter dialogue'}, G_x)$;
2. $\forall p_j = tell(a_j, b_j, L_j, A_j) \in \mathcal{D}$:
   (*i*) *if* $L_j =$ *'state circumstances' then*

$$\mathcal{S}^{a_j} \vDash_{\Delta} G_x \cup \Psi^{sc}_k \cup \Psi^{sa}_k$$

   for some $k \leq j$*, and* $argument(p_j) = \Delta \setminus actions(\Delta)$;
   (*ii*) *if* $L_j =$ *'state actions' then*

$$\mathcal{S}^{a_j} \vDash_{\Delta} G_x \cup \Psi^{sc}_j \cup \Psi^{sa}_j$$

   and $argument(p_j) = actions(\Delta)$;
   (*iii*) *if* $L_j =$ *'deny circumstances' then*

$$\exists \Psi^{sc} \subseteq \Psi^{sc}_j, \Psi^{sa} \subseteq \Psi^{sa}_j, h \in \Psi^{sc}_j \setminus \Psi^{sc}$$

   such that $\mathcal{S}^{a_j} \cup \Psi^{sc} \cup \Psi^{sa} \vDash_{\Delta} h' \cup G_x$ *and* h' *attacks* h*, and* $argument(p_j) = h$;
   (*iv*) *if* $L_j =$ *'deny actions' then*

$$\exists \Psi^{sc} \subseteq \Psi^{sc}_j, \Psi^{sa} \subseteq \Psi^{sa}_j, h \in \Psi^{sa}_j \setminus \Psi^{sa}$$

   such that $\mathcal{S}^{a_j} \cup \Psi^{sc} \cup \Psi^{sa} \vDash_{\Delta} h' \cup G_x$ *and* h' *attacks* h*, and* $argument(p_j) = h$;
   (*v*) *in all other cases, except for* $L_j =$ *'enter dialogue',* $argument(p_j) = \emptyset$.
3. $\nexists p_j, p_k \in \mathcal{D}$ *such that* $p_j = p_k \ \wedge \ j \neq k$,

where for a given set Δ, $actions(\Delta) = \{\mathbf{E}(a) \in \Delta \text{ such that } a \in Actions\}$*. We will call* x *the* initiator *of* $\mathcal{D}$.

Thus, in an argumentation dialogue, the agents focus on a specific goal (1). They do not exchange purely "dialogical" arguments, but genuine products of their own reasoning based on the knowledge available to them. In particular, we require that circumstances/actions stated are supported by the uttering agent (2-*i*/*ii*), and for those denied the agent is able to produce an attacking argument based on the goal subject of the dialogue (2-*iii*/*iv*). Finally, we require that an agent does not utter the same performative twice (3). In this way, at each step j, the dialogue develops by an agent reasoning on the state at step k, for some $k < j$, to propose a new state to the receiver. Dialogue moves need not directly

address the previous move, but are free to refer to moves uttered in the past, in the course of the same dialogue. This leaves agents free to try several alternative arguments, so that the dialogue can proceed even if an agent does not have an answer to the last move.

One can easily see that, if $\mathcal{L}_{arg}$ is composed of a finite number of ground arguments, because of condition 3 of Definition 14 ($\nexists p_j, p_k \in \mathcal{D}$ such that $p_j = p_k \ \wedge \ j \neq k$), dialogues will always finite length [19]. However, we are interested here not only in dialogues that terminate, but especially we want to be able to define what dialogues are "fruitful." We will then focus on the notion of agreement:

Definition 15 (Agreement between two agents). *Given a multiagent argumentation framework $\mathcal{M}$, an* agreement *between two agents $x, y \in \mathcal{M}$ about a goal G_x is a set $\mathcal{C}$ such that there exists an* argumentation dialogue *$\mathcal{D} = \{p_0, p_1, \ldots\}$ between x and y about G_x, whose state(D, j) is such that $\Psi_j^{aa} = \mathcal{C}$ for some j.*

In other words, we say that two agents reach an agreement when they come up in the course of the same dialogue with a set $\mathcal{C}$ which contains the same actions. By definition of argumentation dialogue, they are supported by the same arguments (circumstances) from both sides.

This formulation of argumentation dialogue makes it possible to prove some important properties of the framework, which to the best of our knowledge are not to be found in other multiagent argumentation frameworks.

Proposition 1. *Given an argumentation dialogue $\mathcal{D}$ and a performative $p \in \mathcal{D}$, argument(p) is a conflict-free set of arguments.*

Proof. By Definition 14, $\forall p \, \exists \, \Delta, \mathcal{S},$ and $\mathcal{G}$ such that $\mathcal{S} \vDash_{\Delta} \mathcal{G}$ and *argument*$(p) \subseteq \Delta$. Thus Proposition 1 follows from Corollary 2.

Proposition 2. *Given an argumentation dialogue $\mathcal{D}$ and a performative $p \in \mathcal{D}$:*

1. *if locution$(p) \in \{$'state circumstances', 'state actions'$\}$, then argument(p) is an admissible set of arguments for utterer(p);*
2. *if locution$(p) \in \{$'deny circumstances', 'deny actions'$\}$, then argument(p) is an admissible set of arguments for receiver(p);*

Proof. The proof follows from Corollary 3 and from Definition 14.

Proposition 3. *Every agreement $\mathcal{C}$ between two agents x and y about a goal G_x, is an admissible set of arguments for both x and y.*

Proof. If $\mathcal{C} = \emptyset$, Proposition 3 follows from Definition 11, which implies that the empty set is always admissible. If $\mathcal{C} \neq \emptyset$, by Definition 15 there exists an *argumentation dialogue* $\mathcal{D} = \{p_0, p_1, \ldots\}$ between x and y about G_x, whose *state*(D, j) is such that $\Psi_j^{aa} = \mathcal{C}$ for some j. By Corollary 5 it is a structural property of *state*$(\mathcal{D}, j)$ that $\Psi_j^{aa} = \Psi_j^{sa}$, and by Definition 13 $\exists l, k$ such that $\Psi_j^{sa} =$

$argument(p_l) = argument(p_k)$ and $locution(p_l) = locution(p_k) = `state\ actions'$ for some $l, k \leq j$. It thus follows from Proposition 2 that Ψ_j^{sa} is admissible for both x and y.

We believe that this is a very important property of $\mathcal{S}$CIFF-AF. If two agents reach what we call an agreement during a dialogue, it is important that such an agreement identifies a possible future system development which is admissible by both – which is the case here. In this way, agents can step through agreements and thus develop plans for future courses of action which ensure a consistent system evolution.

4 An example of an argumentation dialogue leading to an agreement using SCIFF-AF

In order to illustrate the usage of the $\mathcal{S}$CIFF-AF framework and its properties, we propose as a scenario an adaptation of Rahwan & Amgoud's conference example [15]:

Example 1. A scientist s (based in the UK) wishes to attend a $conf$erence. Prior to his departure, however, he needs to reach a preliminary agreement with his department d. s knows that $conf$ is in Liverpool, and that the fee can be 400 (on-site) or 200 (early), that a limo is a comfortable car, and that Liverpool is a far but domestic destination. s has some constraints: he knows that if he wishes to attend a conference, then he must reach the place of the conference, and pay the fee. If he wishes to reach a place, he must either fly or drive. In addition, if he wishes to reach a place, either it is not a domestic destination, and he does not want to fly economy, nor he wants to drive; or it is a far destination, and in that case he does not want to drive; or else he wants to rent a comfortable car. s's department, d, has a number of constraints. If one wants to reach a destination and pay a conference fee, then he must attend the conference; the fee must be lower than 300, or else it is not permitted to rent a limo, nor to fly business, or else it is a domestic destination, and then it is not permitted to fly business.

Given such a scenario, a possible argumentation dialogue that we would like to obtain in this framework could be the following:

1. (s): I wish to attend a conference ($conf$).
2. (d): I am listening.
3. (s): There are some circumstances I wish to bring to your attention. I do not want to drive there. So I will not rent a car. Also, I think I have to pay on-site registration.
4. (s): I was thinking I can do the following: buy a business plane ticket, fly and reach Liverpool, pay 400 as a fee.
5. (d): You are not allowed to fly business class!
6. (s): I take your point.
7. (s): I can fly economy.

$\mathcal{A}$ $\{early, on\_site\} \cup \mathbf{EXP}$

$Actions$ $\{reach, fly, drive, pay, buy\_ticket, rent\_car\}$

$\mathcal{G}_s$ $\{\mathbf{E}(attend(conf))\}$

$\mathcal{IC}_s$ $\mathbf{E}(attend(Conf)) \rightarrow conference(Conf, Venue, Fee) \ \wedge \ \mathbf{E}(reach(Venue))$
$\wedge \ \mathbf{E}(pay(Conf, Fee))$.
$\mathbf{E}(reach(Dest)) \rightarrow \mathbf{E}(fly(Dest)) \ \vee \ \mathbf{E}(drive(Dest))$.
$\mathbf{E}(reach(Dest)) \rightarrow non\,domestic(Dest) \ \wedge \ \mathbf{EN}(buy\_ticket(Dest, economy))$
$\wedge \ \mathbf{EN}(drive(Dest))$
$\vee \ far(Dest) \ \wedge \ \mathbf{EN}(drive(Dest))$
$\vee \ domestic(Dest) \ \wedge \ \mathbf{E}(rent\_car(Dest, Car)) \ \wedge \ comfortable(Car)$.
$\mathbf{E}(fly(Dest)) \rightarrow \mathbf{E}(buy\_ticket(Dest, economy)) \ \vee \ \mathbf{E}(buy\_ticket(Dest, business))$.
$\mathbf{E}(drive(Dest)) \rightarrow \mathbf{E}(rent\_car(Dest, sedan)) \ \vee \ \mathbf{E}(rent\_car(Dest, limo))$.
$\mathbf{E}(fly(Dest)) \ \wedge \ \mathbf{E}(drive(Dest)) \rightarrow \bot$.
$\mathbf{E}(fly(Dest)) \rightarrow \mathbf{EN}(rent\_car(Dest, Car))$.
$\mathbf{E}(drive(Dest)) \rightarrow \mathbf{EN}(buy\_ticket(Dest, Class))$.
$early \ \wedge \ on\_site \rightarrow \bot$.

$\mathcal{P}_s$ $conference(conf, lvp, Fee) \leftarrow (on\_site \ \wedge \ Fee = 400) \ \vee \ (early \ \wedge \ Fee = 200)$.
$comfortable(limo)$.
$far(lvp)$.
$domestic(lvp)$.

$\mathcal{IC}_d$ $\mathbf{E}(reach(Dest)) \ \wedge \ \mathbf{E}(pay(Conf, Fee)) \rightarrow \mathbf{E}(attend(Conf)) \ \wedge \ Fee < 300$
$\vee \ \mathbf{EN}(rent\_car(Dest, limo)) \ \wedge \ \mathbf{EN}(buy\_ticket(Dest, business))$
$\wedge \ domestic(Dest) \ \vee \ \mathbf{EN}(buy\_ticket(Dest, business))$.
$\mathbf{E}(buy\_ticket(Dest, business)) \ \wedge \ \mathbf{E}(buy\_ticket(Dest, economy)) \rightarrow \bot$.

$\mathcal{P}_d$ $domestic(lvp)$.

Fig. 1. $\mathcal{S}$CIFF programs of scientist (s) and department (d)

8. (d): Agreed. Go ahead.

Figure 1 shows its possible implementation in the $\mathcal{S}$CIFF-AF. Note that, in addition to its formulation given above, some additional domain-specific constraints are specified: in order to fly one must either buy an economy ticket or a business ticket, one does never want to fly and drive at the same time, etc. Note also that s does not know which one is the fee he has to pay, so it considers *early* and *on_site* to be an abducible atoms belonging to $\mathcal{A}$. s's goal, ICs and program are denoted by $\mathcal{G}_s$, $\mathcal{IC}_s$ and $\mathcal{P}_s$; similarly for d.

The agents can engage in argumentation dialogues to find a possible future evolution upon which both agree. One such dialogue is shown in Figure 2. At each step, the dialogue complies with Definition 14, and it therefore produces a result which (i) is consistent with both constraints, and (ii) is such that in the end both agree on the present/future circumstances. In fact, the dialogue ends with an agreement ($Actions_2$).

Thanks to the result enunciated in Proposition 3, we know that $Actions_2$ is indeed an admissible set of arguments for both s and d.

$p_0 : tell(s, d, `enter\ dialogue', \{\mathbf{E}(attend(conf))\}).$
$p_1 : tell(s, d, `turn\ finished').$
$p_2 : tell(d, s, `enter\ dialogue', \{\mathbf{E}(attend(conf))\}).$
$p_3 : tell(d, s, `turn\ finished').$
$p_4 : tell(s, d, `state\ circumstances', Terms_1).$
$p_5 : tell(s, d, `state\ actions', Actions_1).$
$p_6 : tell(s, d, `turn\ finished').$
$p_7 : tell(d, s,' deny\ actions', \mathbf{E}(buy\_ticket(lvp, business)))$
$p_8 : tell(d, s, `turn\ finished')$
$p_9 : tell(s, d,' accept\ denial')$
$p_{10} : tell(s, d, `state\ circumstances', Terms_2).$
$p_{11} : tell(s, d, `state\ actions', Actions_2).$
$p_{12} : tell(s, d, `turn\ finished')$
$p_{13} : tell(d, s,' state\ actions', Action_2)$
$p_{14} : tell(d, s, `turn\ finished')$
$p_{15} : tell(s, d,' leave\ dialogue')$
$p_{16} : tell(d, s,' leave\ dialogue')$
$Terms_1 = \{\ on\_site, not\ early, \mathbf{EN}(rent\_car(lvp, Car)), \mathbf{EN}(drive(lvp)), not\ \mathbf{E}(drive(lvp))\ \}$
$Actions_1 = \{\ \mathbf{E}(reach(lvp)), \mathbf{E}(pay(conf, 400)), \mathbf{E}(buy\_ticket(lvp, business)), \mathbf{E}(fly(lvp))\ \}$
$Terms_2 = \{\ on\_site, not\ early, \mathbf{EN}(buy\_ticket(lvp, business)), \mathbf{EN}(rent\_car(lvp, Car)),$
$\mathbf{EN}(drive(lvp)), not\ \mathbf{E}(drive(lvp)), \mathbf{EN}(drive(lvp)), not\ \mathbf{E}(drive(lvp))\ \}$
$Actions_2 = \{\ \mathbf{E}(reach(lvp)), \mathbf{E}(pay(conf, 400)), \mathbf{E}(buy\_ticket(lvp, economy)), \mathbf{E}(fly(lvp))\ \}$

Fig. 2. Sample argumentation dialogue between s and d about $G_s = \mathbf{E}(attend(conf))$

5 Conclusion and Future Work

The main contribution of this paper is the illustration of some fundamental properties of a declarative framework for multiagent reasoning and dialogue-based argumentation about actions ($\mathcal{S}$CIFF-AF), initially proposed in [20].

$\mathcal{S}$CIFF-AF is equipped with a sound operational model, an admissible sets semantics, a notion of (argumentation) dialogue and a notion of agreement about actions. Thanks to these properties, it is possible to accommodate in $\mathcal{S}$CIFF-AF a declarative representation of the agent knowledge, upon which agents can reason, and interact by argumentation dialogues.

Although agent reasoning is not covered by this work, Alberti *et al.* have proposed in [1] an agent architecture in which the reasoning activity of agents is based on $\mathcal{S}$CIFF, so we have ground to believe that $\mathcal{S}$CIFF-AF can be actually used as a concrete, operational multiagent argumentation framework.

The operational nature of $\mathcal{S}$CIFF-AF is maybe one of its main distinguishing features, compare to other existing work. Argumentation dialogues are useful because through them agents may eventually reach mutual agreements, which they can directly use, for example by adopting them as possible future internal goals. Importantly, in this article we have demonstrated that $\mathcal{S}$CIFF-AF is grounded on a solid formal basis, which includes a number of results about its relation with Dung's abstract argumentation framework.

type of goal	type of expectation
positive	**E**
negative	**EN**
in abeyance	*not* **E** $\wedge$ *not* **EN**

Fig. 3. Mapping between types of goals and types of expectations

This work builds on previous results on abstract argumentation frameworks [7], on the $\mathcal{S}$CIFF proof-procedure [2], on computing arguments in ALP [12], and on multi-agent dialogue framework [18, 5], as cited in the text. In the future, we intend to investigate more thoroughly the formal relations between $\mathcal{S}$CIFF-AF and other argumentation frameworks. In particular, we intend to focus on Amgoud & Kaci's work [3], in which goals are partitioned into three categories: *positive* goals, *negative* goals, and goals in *abeyance*. If positive goals reward the agent that satisfies them, negative goals are on the contrary those considered unacceptable, while goals in abeyance just mirror what is not rejected, although they do not really reward the agent that adopts them. We think that the $\mathcal{S}$CIFF-AF metaphor of expectations applies smoothly to this understanding of goals. One obvious relation among the two paradigms is shown in Figure 3.

Beside the very similar understanding of goals/expectations, Amgoud & Kaci's framework and its recent refinement by Rahwan & Amgoud [15] do have many motivations in common with this work. We plan to investigate these aspects in depth in the future. Some possible interesting extensions of the $\mathcal{S}$CIFF-AF framework could be a notion of attack that accommodates a *priority degree*, and a more comprehensive argumentation setting in which agents argue using not only atomic entities, but also implications (i.e., integrity constraints, or conditional rules). Another aspect worth investigating is that of knowledge representation, for example to distinguish between *explanatory arguments*, used to provide reasons of adopting goals, beliefs or disbeliefs, and *instrumental arguments*, used to present plans to achieve goals [3, 15].

Acknowledgements

The author thanks Federico Chesani, Marco Gavanelli, Francesca Toni, the anonymous referees and the ArgNMR 2007 workshop participants[2] for fruitful discussion and valuable feedback, suggestions and comments on earlier versions of this work. This research has been partially supported by the MIUR PRIN 2005 project No 2005-011293, *Specification and verification of agent interaction protocols*,[3] and by the National FIRB project *TOCAI.IT*.[4]

[2] `http://lia.deis.unibo.it/confs/argnmr`

[3] `http://www.ricercaitaliana.it/prin/dettaglio_completo_prin_en-2005011293.htm`

[4] `http://www.dis.uniroma1.it/~tocai/`

References

1. M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, and P. Mello. A verifiable logic-based agent architecture. In *Proc. ISMIS 2006*, LNCS 2870:338–343. Springer-Verlag.
2. M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Verifiable agent interaction in abductive logic programming: the SCIFF framework. *ACM Transactions on Computational Logic (ToCL)*, accepted, 2007.
3. L. Amgoud and S. Kaci. On the generation of bipolar goals in argumentation-based negotiation. In *ArgMAS 2004*, LNAI 3366:192–207. Springer, 2005.
4. L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue and negotiation. In *Proc. ECAI 2000*, pp. 338–342. IOS Press.
5. K. Atkinson, T. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument over proposals for action. *Journal of Autonomous Agents and Multi-Agent Systems*, 11:153–171, 2005.
6. S. Coste-Marquis, C. Devred, and P. Marquis. Symmetric argumentation frameworks. In *Proc. ECSQUARU*, LNCS 3571:317–328. Springer, 2005.
7. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
8. T. H. Fung and R. A. Kowalski. The IFF proof procedure for abductive logic programming. *Journal of Logic Programming*, 33(2):151–165, 1997.
9. M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *Proc. AAAI'87*, pp. 677–682. Morgan Kaufmann.
10. A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive Logic Programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.
11. A. C. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *Proc. AAMAS 2003*, pp. 883–890. ACM Press.
12. A. C. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, 9(4):515–562, 1999.
13. J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 2nd extended edition, 1987.
14. P. McBurney, S. Parsons, and M. Wooldridge. Desiderata for agent argumentation protocols. In *Proc. AAMAS 2002*, pp. 402–409. ACM Press.
15. I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. In *Proc. AAMAS 2006*, pp. 347–354. ACM Press.
16. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18:343–375, 2003.
17. A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In *Proc. KR&R '91*, pp. 473–484. Morgan Kaufmann.
18. F. Sadri, F. Toni, and P. Torroni. Logic agents, dialogues and negotiation: an abductive approach. In *Proc. AISB'01 Convention, York, UK*, March 2001.
19. P. Torroni. A study on the termination of negotiation dialogues. In *Proc. AAMAS 2002*, pp. 1223–1230. ACM Press.
20. P. Torroni. Multi-agent agreements about actions through argumentation. In *Computational Models of Argument*, Vol. 144 of *Frontiers in Artificial Intelligence and Application*, pp. 323–328. IOS Press, 2006.