Classification and Strategical Issues of Argumentation Games on Structured Argumentation Frameworks

Matthias Thimm
Technische Universität Dortmund
Germany
matthias.thimm@tu-dortmund.de

Alejandro J. García
Universidad Nacional del Sur, Bahía Blanca
Argentina
ajg@cs.uns.edu.ar

ABSTRACT

This paper aims at giving a classification of argumentation games agents play within a multi-agent setting. We investigate different scenarios of such argumentation games that differ in the protocol used for argumentation, i. e. direct, synchronous, and dialectical argumentation protocols, the awareness that agents have on other agents beliefs, and different settings for the preferences of agents. To this end we employ structured argumentation frameworks, which are an extension to Dung's abstract argumentation frameworks that give a simple inner structure to arguments. We also provide some game theoretical results that characterize a specific argumentation game as strategy-proof and develop some argumentation selection strategies that turn out to be the dominant strategies for other specific argumentation games.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent Agents

General Terms

Theory

Keywords

Argumentation, Game Theory

1. INTRODUCTION

Argumentation has become a popular choice for reasoning with inconsistent information in artificial intelligence [3]. On the one hand, there are a lot of approaches to model argumentation in different kinds of logics, e. g. classical logic [4], defeasible logic [11], or possibilistic logic [1]. On the other hand, *abstract argumentation* as introduced by Dung [6] is widely used as an abstract means to talk about argumentation in general. In abstract argumentation, arguments are represented as atomic entities and the interrelationships between different arguments are modeled using an attack relation. Abstract argumentation has been thoroughly investigated in the past ten years and there is quite a lot of work on, e. g. semantical issues [2] and extensions of abstract argumentation frameworks [9].

In the context of agent and multi-agent systems, there are mainly two applications of formal argumentation. First, using argumentation techniques as a non-monotonic reasoning process within a single agent and second, using argumentation in dialogues between different agents in order to realize persuasion, cooperation, planning, or general conflict solving. Here, we focus on the second application where reasoning is performed involving the whole system of agents, see e. g. [7] for a formalization. In a dialogue, agents take turns in bringing up arguments for some given claim and depending on the interrelationships of the arguments the claim is accepted or rejected by the agents (either individually or jointly). Up until recently, strategic issues in argumentation dialogues have been mostly ignored with few exceptions, e. g. [15]. By considering game theoretical aspects in argumentation dialogues [13] the interest in strategies for the selection of arguments and the general connection of game theory and argumentation has grown. An argumentation dialogue can be represented as a strategic game involving a set of self-interested agents and in choosing the "right" arguments agents can influence the outcome of the argumentation and reach a more desirable result according to their own preferences. In [12, 14, 13] Rahwan et al. investigate a specific argumentation game, in which agents have to state all arguments they wish at once. Because of this simple setting they were able to prove strategy-proofness of this scenario under certain circumstances, i. e. the *dominant strategy* of each agent is to truthfully report all their arguments. Besides this scenario of *direct argumentation* there are other formalizations of specific argumentation games, e. g. [15]. But up till now, to our knowledge there has been no comprehensive overview on the different argumentation settings and the different scenarios where agents can argue with each other.

The contribution of this paper is twofold. The main contribution lies in a classification of the different argumentation games agents can play within a multi-agent setting. We make a first attempt to characterize argumentation games by means of the used game protocol, the awareness of the agents on other agents beliefs, and the structure of the preferences of the agents. We use structured argumentation frameworks, a novel approach which generalizes abstract argumentation frameworks, to model argumentation between different agents. The second contribution lies in general-

Cite as: Classification and Strategical Issues of Argumentation Games on Structured Argumentation Frameworks, Matthias Thimm, Alejandro J. García, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

izing the strategy-proofness result of [12] and investigating several other settings for argumentation games in terms of the strategical issues involving argument selection.

The rest of this paper is organized as follows. In Sec. 2 we give a brief overview on abstract argumentation and introduce the novel approach of structured argumentation. We continue in Sec. 3 with applying structured argumentation onto a multi-agent setting. Section 4 develops a classification of argumentation games in the multi-agent setting in terms of game protocol, awareness, and agent types. We investigate several strategical issues in some instances of argumentation games in Sec. 5 and conclude in Sec. 6.

2. FORMAL ARGUMENTATION

2.1 Abstract Argumentation

Abstract argumentation frameworks [6] take a very simple view on argumentation as they do not presuppose any internal structure of an argument. Abstract argumentation frameworks only consider the interactions of arguments by means of an attack relation between arguments.

Definition 1. An *abstract argumentation framework* AF is a tuple $\mathsf{AF} = (\mathsf{Arg}, \mathsf{attacks})$ where Arg is a set of arguments and $\mathsf{attacks}$ is a relation $\mathsf{attacks} \subseteq \mathsf{Arg} \times \mathsf{Arg}$.

For two arguments $\mathcal{A}, \mathcal{B} \in \mathsf{Arg}$ the relation $(\mathcal{A}, \mathcal{B}) \in \mathsf{attacks}$ means that argument $\mathcal{A}$ attacks argument $\mathcal{B}$. Abstract argumentation frameworks can be concisely represented as directed graphs, where arguments are represented as nodes and edges model the attack relation.

Example 1. Consider the abstract argumentation framework $\mathsf{AF} = (\mathsf{Arg}, \mathsf{attacks})$ depicted in Fig. 1. Here it is $\mathsf{Arg} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$ and $\mathsf{attacks} = \{(\mathcal{A}_1, \mathcal{A}_2), (\mathcal{A}_2, \mathcal{A}_3), (\mathcal{A}_2, \mathcal{A}_4), (\mathcal{A}_3, \mathcal{A}_2), (\mathcal{A}_3, \mathcal{A}_4)\}$.

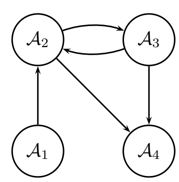

Figure 1: A simple argumentation framework

Semantics are given to abstract argumentation frameworks by means of extensions. An *extension* E of an abstract argumentation framework $\mathsf{AF} = (\mathsf{Arg}, \mathsf{attacks})$ is a set of arguments $E \subseteq \mathsf{Arg}$ that gives some coherent view on the argumentation underlying AF. In the literature [6] a wide variety of different types of extensions has been proposed. All these different types of extensions require some basic properties as *conflict-freeness* and *admissibility.* A set $S \subseteq \mathsf{Arg}$ is *conflict-free* if and only if there are no two arguments $\mathcal{A}, \mathcal{B} \in \mathsf{Arg}$ with $(\mathcal{A}, \mathcal{B}) \in \mathsf{attacks}$. An argument $\mathcal{A} \in \mathsf{Arg}$ is *acceptable* with respect to a set of arguments $S \subseteq \mathsf{Arg}$ if and only if for every argument $\mathcal{B} \in \mathsf{Arg}$ with $(\mathcal{B}, \mathcal{A}) \in \mathsf{attacks}$ there is an argument $\mathcal{C} \in S$ with $(\mathcal{C}, \mathcal{B}) \in \mathsf{attacks}$. A set $S \subseteq \mathsf{Arg}$ is *admissible* if and only if it is conflict-free and every argument $a \in S$ is acceptable with respect to S.

Extensions of an abstract argumentation framework can be described using the characteristic function F_{AF} defined as $F_{\mathsf{AF}}(S) = \{\mathcal{A} \in \mathsf{Arg} \mid \mathcal{A} \text{ is acceptable wrt. } S\}$ for sets $S \subseteq \mathsf{Arg}$.

Definition 2. Let $\mathsf{AF} = (\mathsf{Arg}, \mathsf{attacks})$ be an abstract argumentation framework and $S \subseteq \mathsf{Arg}$ an admissible set.

- S is a *complete extension* if and only if $S = F_{\mathsf{AF}}(S)$.
- S is a *grounded extension* if and only if it is a minimal complete extension (with respect to set inclusion).
- S is a *preferred extension* if and only if it is a maximal complete extension (with respect to set inclusion).
- S is a *stable extension* if and only if it is a complete extension and attacks each $\mathcal{A} \in \mathsf{Arg} \setminus S$.

Example 2. We continue Ex. 1. Due to $F_{\mathsf{AF}}(\{\mathcal{A}_1, \mathcal{A}_3\}) = \{\mathcal{A}_1, \mathcal{A}_3\}$ the set $\{\mathcal{A}_1, \mathcal{A}_3\}$ is a complete extension. Furthermore it is the only complete extension and also grounded, preferred, and stable.

Note that the grounded extension is uniquely determined and always exists, cf. [6].

2.2 Structured Argumentation

In the following, we introduce *structured argumentation frameworks* which extend Dung's abstract argumentation frameworks and are a slightly modified variant of dynamic argumentation frameworks [16]. In structured argumentation frameworks arguments are built using a very simple propositional language, so let Prop denote a finite and fixed set of propositions. The basic structure for structured argumentation frameworks are *basic arguments* which represent atomic inference rules by connecting some set of propositions (the support) to another proposition (the claim).

Definition 3. A *basic argument* $\mathcal{A}$ is a tuple $\mathcal{A} = (\{\alpha_1, \ldots, \alpha_n\}, \beta)$ with $\alpha_1, \ldots, \alpha_n, \beta \in \mathsf{Prop}$ and $\beta \notin \{\alpha_1, \ldots, \alpha_n\}$. For a basic argument $\mathcal{A} = (\{\alpha_1, \ldots, \alpha_n\}, \beta)$ we abbreviate $\mathsf{supp}(\mathcal{A}) = \{\alpha_1, \ldots, \alpha_n\}$ (the support of $\mathcal{A}$) and $\mathsf{cl}(\mathcal{A}) = \beta$ (the claim of $\mathcal{A}$).

For the rest of this paper, let U be some fixed and finite set of basic arguments, called the *universal set of basic arguments.* To keep things simple, we assume that U does not contain any *cyclic dependencies*, i. e. there is no infinite sequence $\mathcal{A}_1, \mathcal{A}_2, \ldots \in U$ with $\mathsf{cl}(\mathcal{A}_i) \in \mathsf{supp}(\mathcal{A}_{i+1})$ for all $i > 0$. Together with an attack relation $\rightarrow \subseteq U \times U$ the set of basic arguments form a *structured argumentation framework* (SAF) $\mathfrak{F} = (U, \rightarrow)$.[1]

Example 3. Consider the SAF $\mathfrak{F}_1 = (U, \rightarrow)$ given by

$$\begin{aligned} U = \{\ & \mathcal{A}_1 = (\emptyset, a), \quad \mathcal{A}_2 = (\{a\}, b), \quad \mathcal{A}_3 = (\emptyset, c) \\ & \mathcal{A}_4 = (\emptyset, d), \quad \mathcal{A}_5 = (\{d\}, e), \quad \mathcal{A}_6 = (\{b\}, f) \\ & \mathcal{A}_7 = (\emptyset, g) \quad \} \end{aligned}$$

and

$$\begin{aligned} \rightarrow \ = \{ \quad & (\mathcal{A}_3, \mathcal{A}_2), (\mathcal{A}_2, \mathcal{A}_4), (\mathcal{A}_5, \mathcal{A}_6), \\ & (\mathcal{A}_5, \mathcal{A}_7), (\mathcal{A}_6, \mathcal{A}_7), (\mathcal{A}_7, \mathcal{A}_5) \quad \} \quad . \end{aligned}$$

The rough structure of $\mathfrak{F}_1$ is depicted in Fig. 2, where the attack relation is represented by solid arrows and "support" by dashed arrows. Notice that Fig. 2 does not contain all the information represented by $\mathfrak{F}_1$ as the propositions the arguments relate to have been omitted.

[1] Although structured argumentation frameworks have the same structure as abstract argumentation frameworks, we deliberately use different notations to avoid ambiguity.

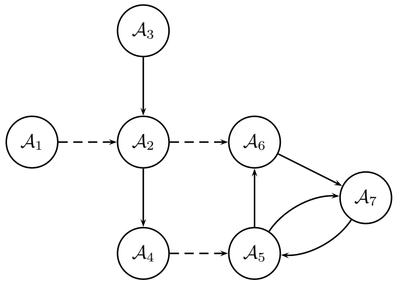

Figure 2: The SAF $\mathfrak{F}_1$

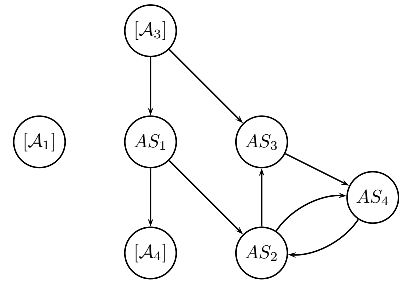

Figure 3: The induced abstract argumentation framework of $\mathfrak{F}_1$ from Ex. 3

A set $S \subseteq U$ is *conflict-free* if and only if there are no two basic arguments $\mathcal{A}, \mathcal{B} \in S$ with $\mathcal{A} \rightarrow \mathcal{B}$. A finite sequence $[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ of basic arguments is conflict-free if and only if $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ is conflict-free. Basic arguments are used to form inference chains called *argument structures*.

Definition 4. Let $S \subseteq U$ be a set of basic arguments and $\mathcal{A} \in S$ a basic argument. An *argument structure* AS for $\mathcal{A}$ with respect to S is a minimal (with respect to set inclusion) conflict-free sequence of basic arguments $AS = [\mathcal{A} = \mathcal{A}_1, \ldots, \mathcal{A}_n]$ with $\{\mathcal{A}_2, \ldots, \mathcal{A}_n\} \subseteq S$ such that for any $\mathcal{A}_i \in AS$ and for any $\alpha \in \mathsf{supp}(\mathcal{A}_i)$ there is a $\mathcal{A}_j \in AS$ with $j > i$ and $\mathsf{cl}(\mathcal{A}_j) = \alpha$ (for $1 \leq i, j \leq n$). Let $\mathsf{ArgStruct}_S(\mathcal{A})$ denote the set of argument structures for $\mathcal{A}$ with respect to S and let $\mathsf{ArgStruct}_S = \bigcup_{\mathcal{A} \in S} \mathsf{ArgStruct}_S(\mathcal{A})$ be the set of all argument structures with respect to S.

For an argument structure $AS = [\mathcal{A}_1, \ldots, \mathcal{A}_n]$ let $\mathsf{top}(AS) = \mathcal{A}_1$ denote the first basic argument in AS. The attack relation $\rightarrow$ on basic arguments can be extended on argument structures by defining $AS_1 \rightarrow AS_2$ if and only if there is a $\mathcal{A} \in AS_2$ with $\mathsf{top}(AS_1) \rightarrow \mathcal{A}$ for two argument structures AS_1 and AS_2. An argument structure AS_1 *indirectly attacks* an argument structure AS_2, denoted by $AS_1 \hookrightarrow AS_2$ if $AS_1 \rightarrow \ldots \rightarrow AS_2$ with an odd number of attacks.

Example 4. We continue Ex. 3. In $\mathfrak{F}_1$ the following sequences are some argument structures

$$\begin{array}{ll} AS_1 = [\mathcal{A}_2, \mathcal{A}_1] & AS_2 = [\mathcal{A}_5, \mathcal{A}_4] \\ AS_3 = [\mathcal{A}_6, \mathcal{A}_2, \mathcal{A}_1] & AS_4 = [\mathcal{A}_7] \end{array}$$

Due to $\mathcal{A}_2 \rightarrow \mathcal{A}_4$ it holds $AS_1 \rightarrow AS_2$. Similarly, it holds $AS_2 \rightarrow AS_3$, $AS_3 \rightarrow AS_4$, $AS_2 \rightarrow AS_4$, $AS_4 \rightarrow AS_2$, and especially $AS_1 \hookrightarrow AS_4$.

Using the extended attack relation, a structured argumentation framework $\mathfrak{F}$ induces an abstract argumentation framework $AF_{\mathfrak{F}} = (\mathsf{Arg}_{\mathfrak{F}}, \mathsf{attacks}_{\mathfrak{F}})$ with $\mathsf{Arg}_{\mathfrak{F}} = \mathsf{ArgStruct}_U$ and $\mathsf{attacks}_{\mathfrak{F}} = \{(AS_1, AS_2) \mid AS_1 \rightarrow AS_2\}$. Let Sem denote one of the Dung-style semantics, cf. Subsec. 2.1. Given a structured argumentation framework $\mathfrak{F}$ and a semantics Sem the *output* of $\mathfrak{F}$ denotes the set of all conclusions acceptable with the semantics Sem in the induced abstract argumentation framework $AF_{\mathfrak{F}}$, cf. [5]. More precisely, if $E_1, \ldots, E_n$ are the extensions of $AF_{\mathfrak{F}}$ under Sem, then $\mathsf{Output}_{Sem}(\mathfrak{F}) = \{\alpha \in \mathsf{Prop} \mid \forall i : \exists AS \in E_i : \mathsf{cl}(\mathsf{top}(AS)) = \alpha\}$.

Example 5. A graphical representation of the induced abstract argumentation framework $AF_{\mathfrak{F}_1}$ of $\mathfrak{F}_1$ from Ex. 3 is depicted in Fig. 3. Note that we abbreviated some argument structures by their names introduced in Ex. 4. The grounded extension E_G of $AF_{\mathfrak{F}_1}$ computes to $E_G = \{[\mathcal{A}_1], [\mathcal{A}_3], [\mathcal{A}_4], AS_2\}$ and therefore $\mathsf{Output}_{\text{grounded}}(\mathfrak{F}_1) = \{a, c, d, e\}$.

Structured argumentation frameworks are a clear generalization of abstract argumentation frameworks as every abstract argumentation framework can be cast into a structured argumentation framework while retaining semantics.

Definition 5. Let $\mathsf{AF} = (\mathsf{Arg}, \mathsf{attacks})$ be an abstract argumentation framework. For every argument $\mathcal{A} \in \mathsf{Arg}$ introduce a new proposition $\mathcal{A} \in \mathsf{Prop}$. The *equivalent structured argumentation framework* $\mathfrak{F}_{\mathsf{AF}} = (U, \rightarrow)$ to AF is defined as

$$\begin{array}{rcl} U & = & \{(\emptyset, \mathcal{A}) \mid \mathcal{A} \in \mathsf{Arg}\} \\ \rightarrow & = & \{((\emptyset, \mathcal{A}), (\emptyset, \mathcal{B})) \mid (\mathcal{A}, \mathcal{B}) \in \mathsf{attacks}\} \end{array}$$

The following theorem states that structured argumentation frameworks are a clear generalization of abstract argumentation frameworks and can easily be verified.

THEOREM 1. *Let AF be an abstract argumentation framework with extensions $E_1, \ldots, E_n$ under some semantics Sem and let $E'_1, \ldots, E'_m$ be the extensions of $\mathsf{AF}_{\mathfrak{F}_{\mathsf{AF}}}$ under Sem. Then there is bijective function $T : \{E_1, \ldots, E_n\} \rightarrow \{E'_1, \ldots, E'_m\}$ such that $T(\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}) = \{(\emptyset, \mathcal{A}_1), \ldots, (\emptyset, \mathcal{A}_k)\}$ for every $E_i = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$, $1 \leq i \leq n$. In particular, it is $n = m$.*

Usually in a multi-agent setting, the universal set of basic arguments U is unknown to all agents because of lack of expertise or just lack of knowledge. When considering a multi-agent setting, every agent may only have a partial view on U and the attack relation. When these agents share their knowledge they dynamically build up a framework which bases on a subset of U.

Definition 6. A *view* $V_{\mathfrak{F}}$ on a structured argumentation framework $\mathfrak{F} = (U, \rightarrow)$ is a structured argumentation framework $V_{\mathfrak{F}} = (U', \rightarrow')$ with $U' \subseteq U$ and $\rightarrow' = \{(\mathcal{A}_1, \mathcal{A}_2) \in \rightarrow \mid \mathcal{A}_1, \mathcal{A}_2 \in U'\}$.

We will omit the subscript of $V_{\mathfrak{F}}$ when the SAF $\mathfrak{F}$ is clear from context.

3. THE MULTI-AGENT SETTING

The scenario we consider can be intuitively described as follows. At the beginning every agent has some view on the underlying SAF $\mathfrak{F}$ and some preferences over the output of the argumentation. The common view considered by all agents as starting point is empty and the agents take turn by bringing up some basic arguments from their own view and incorporating them to the common view. When no agent can bring up more arguments the argumentation ends and an abstract argumentation framework is computed with respect to the final common view. Lastly, this abstract argumentation framework is used to compute the output of the argumentation given some predefined semantics. In the following, we formalize this intuition.

The multi-agent setting is divided into two parts, one describing the basic contents of the scenario, namely the underlying argumentation framework and the agents, and one describing the dynamic part of an evolving argumentation.

Definition 7. A *structured argumentation system* (SAS) Π is a tuple $\Pi = (\mathfrak{F}, Ag)$ with a structured argumentation framework $\mathfrak{F}$ and a set of agent identifiers Ag.

As a simplification we assume that the universal set of basic arguments U of $\mathfrak{F}$ contains exactly the union of the basic arguments appearing in the views of the agents. Hence, any basic argument in U is known by at least one agent. This is not a restriction as an argument not appearing in any view cannot be used at all.

Dynamism is introduced by considering evolving *states* of Π. At any time the state of Π is determined by a current common view V^0, the views of each agent V^i, and the outcome of the argumentation.

Definition 8. A *state* Γ^Π of a SAS $\Pi = (\mathfrak{F}, Ag)$ with $Ag = \{A_1, \ldots, A_n\}$ is a tuple $\Gamma^\Pi = (V^0, \{V^1, \ldots, V^n\}, O)$ with views $V^0, \ldots, V^n$ on $\mathfrak{F}$, and a set $O \subseteq \mathsf{Prop}$. Let Δ^Π denote the set of all states of Π.

We will omit the superscripts Π when Π is clear from context. The final component O of a state Γ denotes the output of the argumentation if Γ is the final state. If the final state has not been reached yet, we set $O = \mathsf{nil}$, where nil is a special identifier denoting no output. For a state $\Gamma = (V^0, \{V^1, \ldots, V^n\}, O)$ we denote $V^i(\Gamma) = V^i$, and $O(\Gamma) = O$. The initial state of a SAS Π is denoted by Γ_0^Π with $O(\Gamma_0^\Pi) = \mathsf{nil}$. The state of a SAS Π evolves over time when agents bring up new basic arguments from their own views. If an agent has to take turn in an argumentation he does so by using its *selection function*. Given a common view of a SAF and an agent's own view a selection function selects a set of basic arguments of the agent's view to come up with. Let $\mathfrak{P}(S)$ denote the power set of a set S.

Definition 9. Let A_k be an agent identifier. A *selection function* sel^{A_k} for A_k is a function $\mathsf{sel}^{A_k} : \Delta \to \mathfrak{P}(U)$ such that $\mathsf{sel}^{A_k}(\Gamma) \subseteq (U^k \setminus U^0)$ for any $\Gamma \in \Delta$ with $V^k(\Gamma) = (U^k, \to^k)$ and $V^0(\Gamma) = (U^0, \to^0)$.

The condition $\mathsf{sel}^{A_k}(\Gamma) \subseteq (U^k \setminus U^0)$ ensures that the agent brings up new basic arguments that are not already part of the common view. Notice also that an agent may bring up no basic arguments at all via $\mathsf{sel}^{A_k}(\Gamma) = \emptyset$. Intuitively spoken, a selection function implements the strategy of an agent in an argumentation in a game theoretical sense. In game theory, the *performance* of an agent's strategy is evaluated by using the agent's preferences on the outcomes of a game. As in our framework the outcome of the argumentation game is determined by the output of the final common view of the underlying $\mathfrak{F}$ the agent's utility is determined by its utility function which maps sets of propositions, i. e. possible outcomes, to natural numbers, thus describing a ranking on the output.

Definition 10. An *utility function* util^A for an agent identifier A is a function $\mathsf{util}^A : \mathfrak{P}(\mathsf{Prop}) \to \mathbb{N}$.

An agent A with a utility function util^A prefers the outcome (i. e. the output) $L_1 \subseteq \mathsf{Prop}$ over $L_2 \subseteq \mathsf{Prop}$ if $\mathsf{util}^A(L_1) > \mathsf{util}^A(L_2)$.

Other agents observe new basic arguments and integrate these in their own views respectively. As a convenience we abbreviate this operation as follows.

Definition 11. Let $V = (U', \to')$ be a view on $\mathfrak{F} = (U, \to)$ and $\mathcal{A} \in U$ a basic argument. The *view update* of V with $\mathcal{A}$ is a view $V' = V \otimes \mathcal{A}$ on $\mathfrak{F}$ with $V' = (U'', \to'')$ defined as $U'' = U' \cup \{\mathcal{A}\}$ and

$$\to'' \quad = \quad \to' \cup \{(\mathcal{A}, \mathcal{B}) \in \to \mid \mathcal{B} \in U'\} \cup \{(\mathcal{B}, \mathcal{A}) \in \to \mid \mathcal{B} \in U'\}$$

Definition 11 suggests that agents are fully aware of attacks between known arguments. This means that when agents incorporate new basic arguments into their view, all attacks between this argument and arguments already known are incorporated as well. For a set of basic arguments $\mathfrak{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\} \subseteq U$ we define $V \otimes \mathfrak{A} = (\ldots((V \otimes \mathcal{A}_1) \otimes \mathcal{A}_2) \otimes \ldots) \otimes \mathcal{A}_n$.

4. ARGUMENTATION GAMES

The type of argument game that agents play directly influences the strategies agents should use in order to obtain the best outcomes. In [12] Rahwan and Larson investigate mechanism design techniques [8] in order to determine suitable mechanisms, i. e. types of games, for abstract argumentation. For a special case of mechanism and a special type of agents they were able to identify this scenario as a strategy-proof game. As such, the best strategy for the agents is to be truthful about their views and bring up all arguments they know of. Although we obtain a similar result as in [12] of strategy-proofness for a special scenario of a SAS we also have a look on strategies for non-strategy-proof games, cf. Sec. 5.

In this section we give an overview on different settings for argumentation games. To this end we identify three key parameters as follows.

1. *Game protocol*: How do agents take turn and when does the game terminates?
2. *Awareness*: Does an agent have knowledge on the views of other agents?
3. *Agent types*: How are the preferences of an agent organized?

In general we assume for all scenarios that every action undertaken by any agent is recorded by all other agents, cf. the notion of *perfect information* in [15].

4.1 Game Protocol

A protocol describes the extensional rules of an argumentation game and prescribes how agents take turns and which actions can be undertaken. More formally, we describe argumentation game protocols by means of state transition rules as in operational semantics [10] that transform one state of a SAS Π into a new one. Given a SAS Π and some initial state Γ_0^Π of Π the rules of a protocol P are applied to Γ_0^Π and its successor state until a *final state* $\mathsf{final}_P(\Gamma_0^\Pi)$ with $O(\mathsf{final}_P(\Gamma_0^\Pi)) \neq \mathsf{nil}$ is reached. Note that $\mathsf{final}_P(\Gamma_0^\Pi)$ is uniquely determined as we do not allow probabilistic decisions in the agents' strategies. For an agent A its *gain* for Γ_0^Π and P is defined as $\mathsf{gain}_A^P(\Gamma_0^\Pi) = \mathsf{util}_A(O(\mathsf{final}_P(\Gamma_0^\Pi)))$, i.e. the agents utility for the outcome of the argumentation. In the following, let Π be a SAS with $\Pi = (\mathfrak{F}, \{A_1, \ldots, A_n\})$ and Γ a state.

4.1.1 Direct Argumentation Mechanism

A direct argumentation mechanism [12] allows only one single step in the argumentation game. Every agent may put forth any set of basic arguments at once. After this, the mechanism terminates. This can be realized with the single state transition rule T_1^d defined as follows.

$$[T_1^d]\frac{\mathfrak{A} = \mathsf{sel}^{A_1}(\Gamma) \cup \ldots \cup \mathsf{sel}^{A_n}(\Gamma)}{\Gamma \longrightarrow (V^{0'}, \{V^{1'}, \ldots, V^{n'}\}, \mathsf{Output}_{Sem}(V^{0'}))}$$

$$\text{with:} \quad V^{i'} = V^i(\Gamma) \otimes \mathfrak{A} \quad (0 \leq i \leq n)$$

Obviously, the direct argumentation protocol $P^d = \{T_1^d\}$ always terminates after one execution step.

4.1.2 Synchronous Argumentation Mechanism

A generalization of the direct argumentation mechanism is the *synchronous argumentation mechanism.* There, every agent may bring up a set of basic arguments at the same time but the process is repeated until no agent wants to bring up any more basic arguments. There are two variants of this mechanism, one where agents are allowed to bring up new basic arguments even if they have not done so in a previous step, and one where agents cannot bring up any new basic arguments if they previously decided not to do so. We call the second variant a *rigid protocol.* When using a rigid protocol, agents have to carefully deliberate whether they choose to not bring forward any arguments, because they do not get any other chance to do so. In this paper due to lack of space, we only consider the non-rigid variant. The rigid protocol will be elaborated in an upcoming extended version of this paper. The non-rigid variant is realized with the following transition rules.

$$[T_1^s]\frac{\mathfrak{A} = \mathsf{sel}^{A_1}(\Gamma) \cup \ldots \cup \mathsf{sel}^{A_n}(\Gamma) \ \text{ and } \ \mathfrak{A} \neq \emptyset}{\Gamma \longrightarrow (V^{0'}, \{V^{1'}, \ldots, V^{n'}\}, \mathsf{nil})}$$

$$\text{with:} \quad V^{i'} = V^i(\Gamma) \otimes \mathfrak{A} \quad (0 \leq i \leq n)$$

$$[T_2^s]\frac{\mathsf{sel}^{A_1}(\Gamma) \cup \ldots \cup \mathsf{sel}^{A_n}(\Gamma) = \emptyset}{\Gamma \longrightarrow (\,\cdot\,,\,\cdot\,, \mathsf{Output}_{Sem}(V^0(\Gamma)))}$$

The *synchronous argumentation protocol* $P^s = \{T_1^s, T_2^s\}$ also clearly terminates after a finite number of steps, because the number of basic arguments is finite.

4.1.3 Dialectical Argumentation Mechanism

In normal dialogues agents usually alternately take turns when bringing up arguments. In general, this can be realized by a *dialectical argumentation mechanism* where we assume some order of the agents and basic arguments can be brought up with respect to this order. As for the synchronous argumentation mechanism two variants are possible with respect to rigidness of the protocol. Anyway, the protocol needs some extra meta information for the states to select the next agent appropriately and we have to ensure that the protocol terminates if no agent wants to bring up new arguments. To this end we introduce some meta information $M = (k_1, k_2) \in \mathbb{N}^2$ such that k_1 is the index of the agent that last took turn and k_2 counts the number of agents that skipped bringing up new basic arguments since the last one that did. For an initial state Γ_0^Π we set $M = (0, 0)$. Then this protocol is realized by the following transition rules.

$$[T_1^t]\frac{k_2 < n \ \text{ and } \ \mathfrak{A} = \mathsf{sel}^{A_{k_1'}}(\Gamma)}{\begin{array}{lcl}\Gamma & \longrightarrow & (V^{0'}, \{V^{1'}, \ldots, V^{n'}\}, \mathsf{nil}) \\ M = (k_1, k_2) & \longrightarrow & M' = (k_1', k_2')\end{array}}$$

$$\begin{array}{lrcl} \text{with:} & V^{i'} & = & V^i(\Gamma) \otimes \mathfrak{A} \quad (0 \leq i \leq n) \\ & k_1' & = & (k_1 \bmod n) + 1 \\ & k_2' & = & \begin{cases} 0 & \text{if } \mathfrak{A} \neq \emptyset \\ k_2 + 1 & \text{otherwise} \end{cases} \end{array}$$

$$[T_2^t]\frac{k_2 = n}{\begin{array}{lcl}\Gamma & \longrightarrow & (\,\cdot\,,\,\cdot\,, \mathsf{Output}_{Sem}(V^0)) \\ M = (k_1, k_2) & \longrightarrow & M\end{array}}$$

As for the synchronous argumentation protocol the termination of the *dialectical argumentation protocol* $P^t = \{T_1^t, T_2^t\}$ is ensured due to the finiteness of the universal set of basic arguments U.

Notice that a variant of the rigid version of the dialectical argumentation mechanism has been previously employed for an argumentation game in [15].

The general protocols described above allow an agent to bring forward an arbitrary number of arguments at any step. For the synchronous and dialectical mechanisms a restricted variant would be allow an agent to bring forward only a single argument at any step. We call such a protocol an *atomic step* protocol. Together with the option of rigidness we obtain each four variants of the synchronous and dialectical mechanisms.

4.2 Awareness

Our definition of selection functions (Def. 9) is quite general as it takes the whole state of the system into account when determining the basic argument that should be brought forward. In particular, a selection function might be heavily influenced by the views of other agents. Usually, an agent does not have complete and accurate knowledge on the subjective views of other agents. One extreme is that an agent has *no awareness* of other agents views. More formally, a selection function sel^{A_k} of an agent $A_k \in Ag$ is *ignorant* if for all $\Gamma_1, \Gamma_2 \in \Delta$ it holds: If $V_0(\Gamma_1) = V_0(\Gamma_2)$ and $V^k(\Gamma_1) = V^k(\Gamma_2)$, then it is $\mathsf{sel}^{A_k}(\Gamma_1) = \mathsf{sel}^{A_k}(\Gamma_2)$. This means that the decision of agent A_k is at any time only dependent on the agent's own view and the common view.

Usually, an agent has some beliefs of the views of other agents. Let $\mathsf{Bel}_{A_k}(A_j, \Gamma)$ the subjective belief of agent A_k on the view of agent A_j in state Γ, i.e. $\mathsf{Bel}_{A_k}(A_j, \Gamma)$ is itself a view. Then, a selection function sel^{A_k} of A_k is *belief-based* if for all $\Gamma_1, \Gamma_2 \in \Delta$ it holds: If $V^0(\Gamma_1) = V^0(\Gamma_2)$ and $V^k(\Gamma_1) = V^k(\Gamma_2)$ and for all $j \neq k$ it is $\mathsf{Bel}_{A_k}(A_j, \Gamma_1) = \mathsf{Bel}_{A_k}(A_j, \Gamma_2)$, then it is $\mathsf{sel}^{A_k}(\Gamma_1) = \mathsf{sel}^{A_k}(\Gamma_2)$. An agent A_k has *full awareness* if his selection function sel^{A_k} is belief-based and $\mathsf{Bel}_{A_k}(A_j, \Gamma) = V^j(\Gamma)$ for every state $\Gamma \in \Delta$ and $j \neq k$.

In between no awareness and full awareness there is a wide range of incomplete and uncertain awareness of other agents' views, but we will not discuss this topic in the current paper.

4.3 Agent Types

Under the term *agent type* we understand in this paper the way the preferences of the agent are organized. The main reason for arguing with other agents is to persuade other agents or to prove some statement. This goal is represented by the agent's utility function which ranks the possible outcomes of the argumentation. In the following we identify some simple utility functions.

The most simple attitude of an agent towards the outcome of an argumentation is the desire to prove a single proposition, no matter what else is proven.

Definition 12. Let $\alpha \in \mathsf{Prop}$. The utility function util_α is called an *indicator utility function* for α if for any $L \subseteq \mathsf{Prop}$ it is $\mathsf{util}_\alpha(L) = 1$ if $\alpha \in L$ and $\mathsf{util}_\alpha(L) = 0$ otherwise.

The choice of 0 and 1 as the only values for the indicator utility function is arbitrary. Any utility function util with $\mathsf{util}(L) = k$ and $\mathsf{util}(L') = l$ for any $L, L' \subseteq \mathsf{Prop}$ with $\alpha \in L$ and $\alpha \notin L'$ for some α can be normalized to an indicator utility function if $k > l$. Note that the definition of indicator utility functions resembles the rationale behind *focal arguments* in [13]. Because of this, if util_α is the utility function of an agent A we call α the *focal element* of A.

The definition of an indicator function can be extended to comprehend for multiple focal elements as follows.

Definition 13. The utility function util_S is called a *multiple indicator utility function* for $S \subseteq \mathsf{Prop}$ if for any $L \subseteq \mathsf{Prop}$ it is $\mathsf{util}_S(L) = 0$ if $S \nsubseteq L$ and $\mathsf{util}_S(L) = 1$ if $S \subseteq L$.

Notice that it holds $\mathsf{util}_{\{\alpha\}} = \mathsf{util}_\alpha$. This general definition does not demand that S has to be "consistent", i. e. there may be argument structures AS_1 resp. AS_2 for some $\alpha \in \mathsf{Prop}$ resp. $\alpha' \in \mathsf{Prop}$ such that $AS_1 \rightarrow AS_2$. Another variant of an agent's preferences can be characterized by a counting utility function which is similar in spirit to the notion of *acceptability maximising preferences* in [12].

Definition 14. Let $S \subseteq \mathsf{Prop}$. The utility function $\mathsf{util}^{\#}_S$ is called a *counting utility function* for S if for any $L \subseteq \mathsf{Prop}$ it is $\mathsf{util}^{\#}_S(L) = |L \cap S|$.

Notice that it holds $\mathsf{util}^{\#}_{\{\alpha\}} = \mathsf{util}_\alpha$. The difference between a counting utility function and a multiple indicator utility function is that for a multiple indicator utility function all focal elements have to be in the output of an argumentation in order to yield a better utility than zero. An agent with a counting utility function tries to prove as many of his focal elements as possible.

In general, there has to be no direct relationship between an agent's view and his utility function. For example, an agent with an indicator utility function util_α may have no basic argument for α in his own view or, more drastically, his view can give reasons to not believe in α. A special form of views are *subjective views* in which an agent's utility function is consistent with its own view.

Definition 15. Let V be a view on $\mathfrak{F}$. V is a *subjective view* on $\mathfrak{F}$ with respect to a utility function util if and only if $\mathsf{util}(\mathsf{Output}_{Sem}(V))$ is a maximum of util.

Furthermore, a view $V = (U', \rightarrow')$ is *globally consistent with respect to a* SAF $\mathfrak{F}$ if there are no two argument structures AS_1, AS_2 in $\mathfrak{F}$ such that $AS_1 \hookrightarrow AS_2$ and $AS_1 \cap U' \neq \emptyset$ and $AS_2 \cap U' \neq \emptyset$. This means that no two basic arguments in V can be used to construct argument structures that are, in any way, inconsistent to one another.

Figure 4 summarizes the different game parameters we investigate in this paper, ordered by their "complexity". Distance from the origin indicates a more demanding setting with respect to the complexity of the strategy for argument selection.

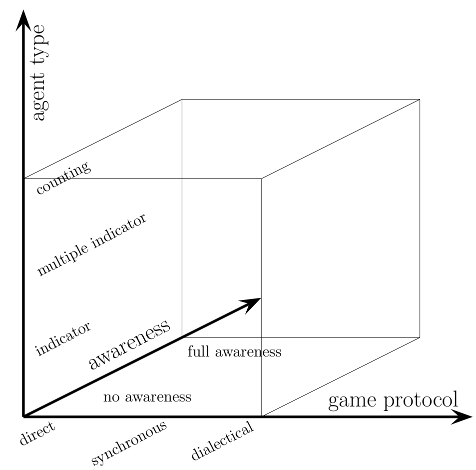

Figure 4: Complexity of game parameters

5. STRATEGIES FOR SELECTING ARGUMENTS

In the following, we investigate some strategies for argument selection in different argumentation games as defined in the previous section. The most simple selection function one can think of is the one that just reports all basic arguments of the agent's view. Let $A_k \in Ag$ be an agent identifier and Γ a state. Then the *truthful selection function* $\mathsf{sel}^{A_k}_{\top}$ is defined as $\mathsf{sel}^{A_k}_{\top}(\Gamma) = U^k \setminus U^0$ with $V^k(\Gamma) = (U^k, \rightarrow^k)$ and $V^0(\Gamma) = (U^0, \rightarrow^0)$. In other words, the selection function $\mathsf{sel}^{A_k}_{\top}$ always returns all basic arguments of an agent's view that aren't already present in the common view of the SAS.

In general, we are interested in finding selection functions that maximize an agent's gain in an argumentation game. Here, an *argumentation game* AG is defined as a tuple $AG = (\Pi, P)$ with a SAS Π and a protocol P. The strongest concept of a selection function that maximizes utility is that of a *dominant strategy.* Let Π be a SAS and let Π' be the same as Π except possibly different selection functions of the agents. Then the selection function sel^{A_k} of agent A_k is a *dominant selection function* if for any such Π' it is $\mathsf{gain}^P_{A_k}(\Gamma^{\Pi}_0) \geq \mathsf{gain}^P_{A_k}(\Gamma^{\Pi'}_0)$. This means, regardless of how the other agents select their arguments, the selection function sel^{A_k} maximizes the gain of agent A_k.[2] The truthful strategy is of special interest in game theory, as it

[2]Notice that agent A_k may have the same selection function sel^{A_k} in Π and Π'.

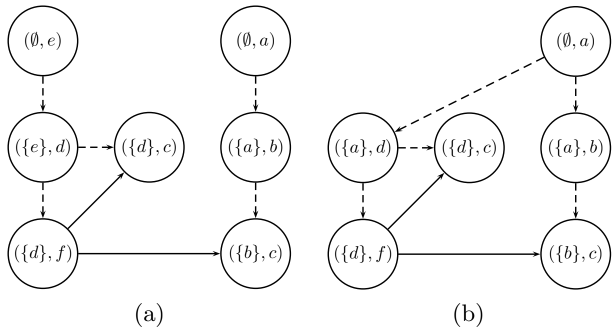

Figure 5: The structured argumentation frameworks (a) $\mathfrak{F}_2$ from Ex. 6 and (b) $\mathfrak{F}_3$ from Ex. 8

is the dominant strategy for *strategy-proof* games. Therefore, given a strategy-proof argumentation game it is the best choice for each agent to truthfully report all their basic arguments. In [12] Rahwan and Larson identified a special type of direct argumentation game as strategy-proof. We can restate and extend their result in our framework as follows.

THEOREM 2. *Let $\Pi = (\mathfrak{F}, Ag)$ be a* SAS. *If the initial view $V^i(\Gamma_0^\Pi)$ of each agent $A_i \in Ag$ is subjective and globally consistent with respect to $\mathfrak{F}$ and the utility function util^{A_i} of each agent A_i is a counting utility function, then (Π, P^d) is strategy-proof.*

Observe that the above statement is independent of the actual chosen semantics due to the general definition of Output. Theorem 2 states that the dominant strategy for subjective and globally consistent views is to use the truthful selection function $\mathsf{sel}_\top$. It is a clear extension of Th. 32 stated in [12] as our underlying argumentation framework is a structured argumentation framework. The statement of Th. 2 easily extends to indicator utility functions, multiple indicator utility functions as well as synchronous and dialectical argumentation protocols. However, the condition of a globally consistent view is hard to check for an agent who has no idea of the structure of the underlying framework $\mathfrak{F}$. Given a basic argument $\mathcal{A}$ in his view he may not know if $\mathcal{A}$ can be used to construct an argument structure against one of his "own" arguments. Due to this observation, Th. 2 is only applicable for an agent if the global consistency is assured by a trustworthy third party or if the agents have full awareness of the other agent's views and thus can verify the global consistency by themselves. Otherwise an agent cannot know if the best strategy is to be truthful.

In general, full awareness is not a realistic assumption in argumentation. When agents cannot verify the global consistency of their view, some strategic deliberations are mandatory as the following example shows.

Example 6. Consider the following SAF $\mathfrak{F}_2 = (U, \rightarrow)$.

$$\begin{aligned} U \;=\; \{ \;& (\emptyset, a), (\{a\}, b), (\{b\}, c), (\emptyset, e), \\ & (\{e\}, d), (\{d\}, f), (\{d\}, c) \;\} \\ \rightarrow \;=\; \{ \;& ((\{d\}, f), (\{d\}, c)), ((\{d\}, f), (\{b\}, c)) \;\} \end{aligned}$$

An overview of $\mathfrak{F}_2$ is given in Fig. 5 (a). Let $\Pi = (\mathfrak{F}_2, \{A_1, A_2\})$ be a SAS and the initial state $\Gamma_0^\Pi = (\emptyset, \{V^1, V^2\}, \mathsf{nil})$ of Π be given as follows.

$$V^1 = (U \setminus \{(\{d\}, f)\}, \cdot) \qquad V^2 = (\{(\{d\}, f)\}, \emptyset)$$

The attack relation of V^1 is omitted but can be determined via Def. 6. Note that view V^1 is subjective but not globally consistent. Imagine A_1 wants to prove c, i. e., the utility function of A_k is util_c. Note that there are two argument structures in $\mathfrak{F}_2$ to prove c while one of them ($[(\{d\}, c), (\{e\}, d), (\emptyset, e)]$) enables A_2 to bring up an attacker, namely $[(\{d\}, f), (\{e\}, d), (\emptyset, e)]$. From a self-interested point of view A_1 should only bring forward the arguments that do not allow A_2 to counterargue.

In the following, we develop some simple strategies for argument selection that generalize the truthful strategy in scenarios where the agent may not have a globally consistent view and that are more cautious in bringing forward arguments. In order to ensure that an agent brings forward only the arguments that are not harmful for proving his focal elements, we define the *attack set* as follows.

Definition 16. Let $\mathfrak{F} = (U, \rightarrow)$ be a SAF and $\alpha \in \mathsf{Prop}$. The *attack set* $\mathsf{AttackSet}_{\mathfrak{F}}(\alpha)$ of α in $\mathfrak{F}$ is defined as

$$\begin{aligned} \mathsf{AttackSet}_{\mathfrak{F}}(\alpha) = \{\, & \mathcal{A} \in U \mid \exists AS_1, AS_2 \in \mathsf{ArgStruct}_U : \\ & \mathcal{A} \in AS_1 \;\wedge\; \mathsf{cl}(\mathsf{top}(AS_2)) = \alpha \;\wedge AS_1 \hookrightarrow AS_2 \quad \} \end{aligned}$$

Intuitively, the set $\mathsf{AttackSet}_{\mathfrak{F}}(\alpha)$ contains all arguments that can be harmful to α in any way. For example, for any argument $\mathcal{A}$ with claim α, the set $\mathsf{AttackSet}_{\mathfrak{F}}(\alpha)$ contains all attackers on $\mathcal{A}$. More generally, $\mathsf{AttackSet}_{\mathfrak{F}}(\alpha)$ contains every argument that belongs to an argument structure that indirectly attacks an argument structure for α. Using attack sets we can define a simple strategy that brings only forward arguments that cannot be harmful in any way.

Definition 17. Let $\alpha \in \mathsf{Prop}$ and A_k an agent identifier. Let $\mathsf{s}^{\mathsf{oc}}_{\alpha, A_k}$ be the selection function defined as

$$\mathsf{s}^{\mathsf{oc}}_{\alpha, A_k}(\Gamma) \quad = \quad \mathsf{sel}_\top^{A_k}(\Gamma) \setminus \mathsf{AttackSet}_{V^k(\Gamma)}(\alpha)$$

for every state Γ. The function $\mathsf{s}^{\mathsf{oc}}_\alpha$ is called the *overcautious selection function* for α.

Although the overcautious strategy is more careful in bringing forward arguments one should note that the determination of $\mathsf{AttackSet}_{V^k(\Gamma)}(\alpha)$ depends on the current view of the agent and might not be complete. The overcautious selection function can be extended to a belief-based selection function by incorporating the beliefs of A_k on the views of the other agents, into the determination of $\mathsf{AttackSet}_{V^k(\Gamma)}(\alpha)$. However, we will not formalize this in the current paper.

Example 7. We continue Ex. 6 but suppose $\mathsf{sel}^{A_1} = \mathsf{s}^{\mathsf{oc}}_{c, A_k}$. Here, A_1 will not bring forward arguments $(\emptyset, e)$ and $(\{e\}, d)$ as they all belong to $\mathsf{AttackSet}_{V_1}(c)$. Note that this strategy is independent of the strategy of any other agent.

Although the overcautious strategy is a very simple strategy for argument selection it is the dominant strategy in a simple class of argumentation games. If an agent has a complete view, i. e., he knows of every argument in the system, but has no awareness on the other agents beliefs, then its best choice is to avoid bringing forward possibly harmful arguments.

THEOREM 3. *Let $\Pi = (\mathfrak{F}, Ag)$ be a* SAS. *For an agent $A_i \in Ag$, if $V_i(\Gamma_0^\Pi) = \mathfrak{F}$ and A_i has no awareness then the overcautious selection function is a dominant strategy for A_i in (Π, P^d).*

The limitations of this simple strategy are reached very quickly as the following small modification of Ex. 6 shows.

Example 8. Consider the following SAF $\mathfrak{F}_3 = (U, \rightarrow)$, cf. Fig. 5 (b).

$$\begin{aligned} U &= \{ (\emptyset, a), (\{a\}, b), (\{b\}, c), (\{a\}, d), (\{d\}, f), (\{d\}, c) \} \\ \rightarrow &= \{ ((\{d\}, f), (\{d\}, c)), ((\{d\}, c), (\{d\}, f)) \} \end{aligned}$$

Let $\Pi = (\mathfrak{F}_3, \{A_1, A_2\})$ be a SAS and $\Gamma_0^\Pi = (\emptyset, \{V_1, V_2\}, \mathsf{nil})$ the initial state of Π with $V^1 = \mathfrak{F}_3$ and $V^2 = (U \setminus \{(\{a\}, d)\}, \cdot)$. Suppose $\mathsf{util}_{A_1} = \mathsf{util}_c$ and $\mathsf{sel}_{A_1} = \mathsf{s}^{\mathsf{oc}}_{c,A_k}$. Here, A_1 will never bring forward argument $(\emptyset, a)$ as $(\emptyset, a) \in \mathsf{AttackSet}_{V_1}(c)$. As a consequence, A_1 will never be able to proof any argument for c.

As Ex. 8 showed it is advisable to bring forward arguments that on the one side may be harmful to one own's desires but on the other side necessary to actually reach the desires. So we refine the overcautious strategy by allowing the agent to bring forward arguments that are inherently necessary for constructing an argument structure for his focal element.

Definition 18. Let $\mathfrak{F} = (U, \rightarrow)$ be a SAF and $\alpha \in \mathsf{Prop}$. The *set of necessary arguments* $\mathsf{NecArg}_{\mathfrak{F}}(\alpha)$ for α in $\mathfrak{F}$ is defined as

$$\mathsf{NecArg}_{\mathfrak{F}}(\alpha) = \bigcap_{\mathcal{A} \in U, \mathsf{cl}(\mathcal{A}) = \alpha, AS \in \mathsf{ArgStruct}_U(\mathcal{A})} AS$$

Definition 19. Let $\alpha \in \mathsf{Prop}$, A_k and agent with a view V and $\mathsf{s}^{\mathsf{c}}_{\alpha,A_k}$ be the selection function defined as

$$\mathsf{s}^{\mathsf{c}}_{\alpha,A_k}(\Gamma) \quad = \quad \mathsf{sel}^{A_k}_{\top}(\Gamma) \setminus (\mathsf{AttackSet}_V(\alpha) \setminus \mathsf{NecArg}_V(\alpha))$$

$\mathsf{s}^{\mathsf{c}}_{\alpha,A_k}$ is called the *cautious selection function* for α.

Example 9. We continue Ex. 8 but suppose $\mathsf{util}_{A_1} = \mathsf{util}_c$ and $\mathsf{sel}_{A_1} = \mathsf{s}^{\mathsf{c}}_{c,A_k}$. Here, A_1 will bring forward argument $(\emptyset, a)$ because it is inherently necessary to construct any argument structure for c.

The cautious strategy performs well in the above example and can be seen as a lower bound for direct argumentation protocols, i. e. the cautious strategy returns as few arguments as necessary.

6. SUMMARY

In this work we have introduced structured argumentation frameworks, a formalism that extends Dung's abstract argumentation frameworks [6] and are a slightly modified variant of dynamic argumentation frameworks [16]. We have used structured argumentation frameworks for defining a multi-agent setting that contains two elements: one describing the basic contents of the scenario, i. e. the underlying argumentation framework and the set of agents; and a second element that describes the dynamic part of an evolving argumentation and determines how the state of the multi-agent system evolves in time. In our framework every agent has its own view on the underlying argumentation framework and its own preferences over the output of the argumentation process. We proposed a first attempt to characterize argumentation games by means of the used game protocol, the awareness of the agents on other agents beliefs, and the structure of the preferences of the agents. We used structured argumentation systems to model argumentation among a group of agents. We have also presented some properties for the proposed framework and protocols.

Acknowledgements. The authors thank the reviewers for their helpful comments to improve the original version of this paper.

7. REFERENCES

[1] T. Alsinet, C. I. Chesñevar, L. Godo, and G. R. Simari. A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems*, pages 1208–1228, 2008.

[2] P. Baroni and M. Giacomin. Skepticism relations for comparing argumentation semantics. *Int. Journal of Approximate Reasoning*, 50(6):854–866, 2009.

[3] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171:619–641, 2007.

[4] P. Besnard and A. Hunter. *Elements of Argumentation.* The MIT Press, 2008.

[5] M. Caminada and L. Amgoud. An axiomatic account of formal argumentation. In *Proceedings of AAAI 2005*, pages 608–613, 2005.

[6] 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):321–358, 1995.

[7] N. C. Karunatillake, N. R. Jennings, I. Rahwan, and P. McBurney. Dialogue games that agents play within a society. *Artificial Intelligence*, 173:935–981, 2009.

[8] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory.* Oxford University Press, 1995.

[9] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9–10):901–934, 2009.

[10] G. D. Plotkin. A structural approach to operational semantics. Technical report, Department of Computer Science, Aarhus University, Aarhus, Denmark, 1981.

[11] H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 219–318. Kluwer, 2002.

[12] I. Rahwan and K. Larson. Mechanism design for abstract argumentation. In *Proceedings of AAMAS 2008*, pages 1031–1038, 2008.

[13] I. Rahwan and K. Larson. Argumentation and game theory. In I. Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 321–339. Springer, 2009.

[14] I. Rahwan, K. Larson, and F. Tohmé. A Characterisation of Strategy-Proofness for Grounded Argumentation Semantics. In *Proceedings of AAAI 2009*, pages 251–256, 2009.

[15] R. Riveret, H. Prakken, A. Rotolo, and G. Sartor. Heuristics in argumentation: A game-theoretical investigation. In *Proceedings of COMMA 2008*, pages 324–335, 2008.

[16] N. D. Rotstein, M. O. Maguillansky, A. J. Garcia, and G. R. Simari. An abstract argumentation framework for handling dynamics. In *Proceedings of NMR 2008*, pages 131–139, 2008.