Mechanism Design for Abstract Argumentation

Iyad Rahwan
[1](Fellow) School of Informatics, University of Edinburgh, Edinburgh EH8 9LE, UK
[2]Faculty of Informatics, British University in Dubai, P.O.Box 502216, Dubai, UAE

Kate Larson
Cheriton School of Computer Science
University of Waterloo
200 University Avenue West, Waterloo
ON, N2L 3G1, Canada

ABSTRACT

Since their introduction by Dung over a decade ago, abstract argumentation frameworks have received increasing interest in artificial intelligence as a convenient model for reasoning about general characteristics of argument. Such a framework consists of a set of arguments and a binary defeat relation among them. Various semantic and computational approaches have been developed to characterise the acceptability of individual arguments in a given argumentation framework. However, little work exists on understanding the strategic aspects of abstract argumentation among self-interested agents. In this paper, we introduce (game-theoretic) argumentation mechanism design (ArgMD), which enables the design and analysis of argumentation mechanisms for self-interested agents. We define the notion of a direct-revelation argumentation mechanism, in which agents must decide which arguments to reveal simultaneously. We then design a particular direct argumentation mechanism and prove that it is strategy proof under specific conditions; that is, the strategy profile in which each agent reveals its arguments truthfully is a dominant strategy equilibrium.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems, coherence and coordination*

General Terms

Theory, Economics

Keywords

Argumentation, Game Theory, Mechanism Design

1. INTRODUCTION

One of the most influential computational models of argument was presented by Dung [3]. Arguments are viewed as abstract entities, with a binary defeat relation among them. This view of argumentation enables high-level analysis while abstracting away from the internal structure of individual arguments. In Dung's approach, given a particular argumentation framework (set of arguments and a binary defeat relation), a rule specifies which arguments should be accepted. A variety of such rules have been studied and compared [1].

However, most research that employs Dung's approach discounts the fact that argumentation is often a multi-agent, adversarial process. Thus, the outcome of argumentation is determined not only by the rules of argument acceptability, but also by the strategies employed by the agents who present these arguments. As these agents may be self-interested, they may have conflicting preferences over which arguments end up being accepted. As such, the design of the argument acceptability rule should take the mechanism design perspective [5, Ch 23]: *what game rules guarantee a desirable social outcome when each self-interested agent selects the best strategy for itself?* Game-theoretic analysis of strategic behaviour in argumentation frameworks is scarce and, to our knowledge, mechanism design has not been applied to multi-agent abstract argumentation to-date.

In this paper, we introduce *argumentation mechanism design* (ArgMD), which enables the design and analysis of argumentation mechanisms for self-interested agents. We recast Dung's abstract framework as a (game-theoretic) mechanism. We then define a particular class of mechanisms, namely *direct* argumentation mechanisms. In these mechanisms, agents must decide which arguments to reveal simultaneously, and the mechanism determines the set of accepted arguments based on some argument acceptability criterion (*e.g.* sceptical, credulous [1]). We then study a particular *sceptical* direct argumentation mechanism, under a certain class of agent preferences (namely, agents want to maximise the number of their own accepted arguments). We examine whether agents have incentive to lie by hiding arguments in an attempt to influence the outcome. We prove that the sceptical mechanism is *strategy-proof*[1] under specific topological restrictions on the argument graph. We further prove that any strategy-proof sceptical argumentation mechanism must satisfy these topological conditions. Thus we have a full characterisation of strategy-proof sceptical argumentation mechanisms, given a certain class of agent preferences.

The paper advances the state-of-the-art in the computational modelling of argumentation in three major ways. Firstly, the paper presents the first definition of the problem of designing argumentation rules in Dung-style frameworks as a (game-theoretic) mechanism design problem. This new

Cite as: Mechanism Design for Abstract Argumentation, Iyad Rahwan and Kate Larson, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

[1] That is, the strategy profile in which each agent reveals its arguments truthfully is a dominant strategy equilibrium.

perspective opens up many possibilities for designing argumentation protocols/rules that have desirable properties in truly adversarial settings. Such perspective on designing argumentation rules has not been possible to-date, and most existing analyses of strategies have been heuristic [6].

The second major contribution of this paper is in demonstrating the power of the ArgMD approach. We provide the first comprehensive game-theoretic analysis of agent strategies in a direct abstract argumentation mechanism. We also characterise conditions under which this mechanism is strategy-proof (truth-revealing), which is the strongest game-theoretic solution concept.

Thirdly, our analysis demonstrates that the properties of argumentation mechanisms depend highly on the form of agent preferences, and on the structure of the argument graph. A variety of other preferences and topological structures are possible, and different ones may be sensible in different application settings. Thus, our work opens new avenues of research for analysing argumentation mechanisms under different conditions.

Next, we present a brief review of abstract argumentation. In Section 3, we present multi-agent abstract argumentation as a mechanism design problem. In Section 4, we define the class of direct abstract argumentation mechanisms and analyse a sceptical instantiation of such mechanisms, showing conditions under which it is strategy-proof. We discuss related work in Section 5 and conclude in Section 6.

2. BACKGROUND

In this section, we briefly outline key elements of abstract argumentation frameworks. We begin with Dung's abstract characterisation of an argumentation system [3]:

Definition 1 (Argumentation framework). *An* argumentation framework *is a pair $AF = \langle \mathcal{A}, \rightharpoonup \rangle$ where $\mathcal{A}$ is a set of arguments and $\rightharpoonup \subseteq \mathcal{A} \times \mathcal{A}$ is a defeat relation. We say that an argument α* defeats *an argument β iff $(\alpha, \beta) \in \rightharpoonup$ (sometimes written $\alpha \rightharpoonup \beta$).*

In this paper, we restrict ourselves to finite argumentation frameworks, that is, frameworks with finite sets of arguments. This assumption is widely adopted in the literature and reflects the reasonable intuition that agents cannot produce *new* relevant information forever. An argumentation framework can be represented as a directed graph (or digraph) in which vertices are arguments and directed arcs characterise defeat among arguments. An example argument graph is shown in Figure 1. Argument α_1 has two defeaters (i.e. counter-arguments) α_2 and α_4, which are themselves defeated by arguments α_3 and α_5 respectively. Different semantics for the notion of argument acceptability have been proposed by Dung [3]. These are stated in the following definitions.

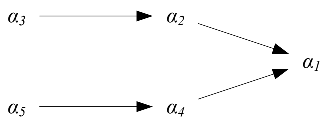


Figure 1: A simple argument graph

Definition 2 (Conflict-free, Defense). *Let $\langle \mathcal{A}, \rightharpoonup \rangle$ be an argumentation framework and let $S \subseteq \mathcal{A}$.*

- *S is* conflict-free *iff there exist no $\alpha \in S$ and $\beta \in S$ such that $\alpha \rightharpoonup \beta$.*
- *S* defends *an argument α iff for each argument $\beta \in \mathcal{A}$, if $\beta \rightharpoonup \alpha$, then there exists an argument $\gamma \in S$ such that $\gamma \rightharpoonup \beta$. We also say that argument α is* acceptable with respect to *S*

Intuitively, a set of arguments is *conflict free* if no argument in that set defeats another. And a set of arguments *defends* a given argument if it defeats all its defeaters.

Example 3. *In Figure 1, $\{\alpha_3, \alpha_5\}$ defends α_1.*

We now look at different semantics that characterise the *collective acceptability* of a set of arguments.

Definition 4 (Characteristic function). *Consider argumentation framework $AF = \langle \mathcal{A}, \rightharpoonup \rangle$. The* characteristic function *of AF is $\mathcal{F}_{AF}$: $2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ such that, given $S \subseteq \mathcal{A}$, we have $\mathcal{F}_{AF}(S) = \{\alpha \in \mathcal{A} \mid S \text{ defends } \alpha\}$.*

When there is no ambiguity about the argumentation framework in question, we will use $\mathcal{F}$ instead of $\mathcal{F}_{AF}$.

Definition 5 (Acceptability semantics). *Let S be a conflict-free set of arguments in framework $\langle \mathcal{A}, \rightharpoonup \rangle$.*

- *S is* admissible *iff it is conflict-free and defends any element in S (i.e. if $S \subseteq \mathcal{F}(S)$).*
- *S is a* complete extension *iff $S = \mathcal{F}(S)$.*
- *S is a* grounded extension *iff it is the minimal (w.r.t. set-inclusion) complete extension (or, alternatively, if S is the least fixed-point of $\mathcal{F}(.)$).*
- *S is a* preferred extension *iff it is a maximal (w.r.t. set-inclusion) complete extension (or, alternatively, if S is a maximal admissible set).*

Let $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ be the set of all possible extensions under a given semantics.

Intuitively, a set of arguments is *admissible* if it is a conflict-free set that defends itself against any defeater –in other words, if it is a conflict free set in which each argument is acceptable with respect to the set itself.

Example 6. *In Figure 1, the sets $\emptyset$, $\{\alpha_3\}$, $\{\alpha_5\}$, and $\{\alpha_3, \alpha_5\}$ are all admissible simply because they do not have any defeaters. The set $\{\alpha_1, \alpha_3, \alpha_5\}$ is also admissible since it defends itself against both defeaters α_2 and α_4.*

An admissible set S is a *complete extension* if and only if *all* arguments defended by S are also in S (that is, if S is a fixed point of the operator $\mathcal{F}$). This captures the attitude of an agent that accepts everything it can defend. There may be more than one complete extension, each corresponding to a particular consistent and self-defending viewpoint.

Example 7. *In Figure 1, the admissible set $\{\alpha_3, \alpha_5\}$ is not a complete extension, since it defends α_1 but does not include α_1. Similarly, sets $\{\alpha_3\}$, $\{\alpha_5\}$ are not complete extensions, since $\mathcal{F}(\{\alpha_3\}) = \{\alpha_3, \alpha_5\}$ and $\mathcal{F}(\{\alpha_5\}) = \{\alpha_3, \alpha_5\}$. The admissible set $\{\alpha_1, \alpha_3, \alpha_5\}$ is the only complete extension, since $\mathcal{F}(\{\alpha_1, \alpha_3, \alpha_5\}) = \{\alpha_1, \alpha_3, \alpha_5\}$.*

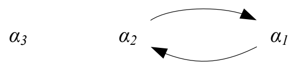


Figure 2: Graph with three complete extensions

As another example, consider the following.

Example 8. *Consider the graph in Figure 2. Here, we have three complete extensions:* $\{\alpha_3\}$*,* $\{\alpha_1, \alpha_3\}$ *and* $\{\alpha_2, \alpha_3\}$*.*

A *grounded extension* contains all the arguments which are not defeated, as well as the arguments which are defended directly or indirectly by non-defeated arguments. This can be seen as a non-committal view (characterised by the *least* fixed point of $\mathcal{F}$). As such, there always exists a unique grounded extension. Dung [3] showed that in finite argumentation systems, the grounded extension can be obtained by an iterative application of the characteristic function to the empty set.

Example 9. *In Figure 1, the grounded extension is* $\{\alpha_1, \alpha_3, \alpha_5\}$*, which is the only complete extension. This can be also calculated using the iterative application of* $\mathcal{F}$ *as follows:*

- $\mathcal{F}^1(\emptyset) = \{\alpha_3, \alpha_5\}$;
- $\mathcal{F}^2(\emptyset) = \mathcal{F}(\mathcal{F}^1(\emptyset)) = \{\alpha_1, \alpha_3, \alpha_5\}$;
- $\mathcal{F}^3(\emptyset) = \mathcal{F}(\mathcal{F}^2(\emptyset)) = \{\alpha_1, \alpha_3, \alpha_5\} = \mathcal{F}^2(\emptyset)$;

Similarly, in Figure 2, the grounded extension is $\{\alpha_3\}$*, which is the minimal complete extension w.r.t set inclusion.*

More intuitively, computing arguments in the grounded extension can be seen as a process of labeling nodes of the defeat graph. First, nodes that have no defeaters are labeled 'undefeated' and the nodes attacked by them are labeled 'defeated.' Then, all labeled arguments are suppressed and the process is repeated on the resulting sub-graph, and so on. If in some iteration, no initial node is found, all unlabeled nodes are labeled as 'defeated' and the process terminates.

A *preferred extension* is a bolder, more committed position that cannot be extended –by accepting more arguments– without causing inconsistency. Thus a preferred extension can be thought of as a maximal consistent set of hypotheses. There may be multiple preferred extensions, and the grounded extension is included in all of them.

Example 10. *In Figure 1,* $\{\alpha_1, \alpha_3, \alpha_5\}$ *is the only preferred extension. But in Figure 2, there are two preferred extensions:* $\{\alpha_1, \alpha_3\}$ *and* $\{\alpha_2, \alpha_3\}$*, which are the maximal complete extension w.r.t. set inclusion.*

Now that the acceptability of sets of arguments is defined, we can define the status of any individual argument.

Definition 11 (Argument status). *Let* $\langle \mathcal{A}, \rightharpoonup \rangle$ *be an argumentation system, and* $\mathcal{E}_1, \ldots, \mathcal{E}_n$ *its extensions under a given semantics. Let* $\alpha \in \mathcal{A}$*.*

1. α *is* sceptically accepted *iff* $\alpha \in \mathcal{E}_i$, $\forall \mathcal{E}_i$ *with* $i = 1, \ldots, n$.
2. α *is* credulously accepted *iff* $\exists \mathcal{E}_i$ *such that* $\alpha \in \mathcal{E}_i$.
3. α *is* rejected *iff* $\nexists \mathcal{E}_i$ *such that* $\alpha \in \mathcal{E}_i$.

An argument is *sceptically accepted* if it belongs to all extensions (equivalently, if it is part of the unique grounded extension). Intuitively, an argument is sceptically accepted if it can be accepted without making any hypotheses beyond what is surely self-defending. On the other hand, an argument is *credulously accepted* on the basis that it belongs to at least one extension. Intuitively, an argument is credulously accepted if there is a possible consistent set of hypotheses in which it is consistent. If an argument is neither sceptically nor credulously accepted, there is no basis for accepting it, and it is therefore *rejected.*

Definition 12 (Accepted Arguments). *Let* $\langle \mathcal{A}, \rightharpoonup \rangle$ *be an argumentation system under semantics* $\mathcal{S}$*. We denote by* $Acc(\langle \mathcal{A}, \rightharpoonup \rangle, \mathcal{S}) \subseteq \mathcal{A}$ *the set of acceptable arguments according to semantics* $\mathcal{S}$*.*

3. ARGUMENTATION AS A MECHANISM DESIGN PROBLEM

The contemporary theory of abstract argumentation frameworks is not concerned with strategic issues in dialogues. In fact, all argument acceptability semantics mentioned in the previous section assume that a set of arguments and a defeat relation are given, and the argument evaluation criteria merely computes the set of acceptable arguments. However, in a multi-agent setting, different arguments are likely to be presented by different self-interested agents. Thus it is crucial to understand the possible strategic behaviour of these agents in terms of what arguments they should or would present. This makes it possible to analyse various argument evaluation criteria in terms of their manipulability. More importantly, understanding strategic behaviour will allow us to devise argument evaluation criteria that ensure that certain desired properties are achieved. To this end, we propose to apply the tools of game theory and mechanism design to abstract argumentation frameworks.

3.1 Game Theory and Mechanism Design

Mechanism design studies the problem of how to ensure that good system-wide decisions or outcomes arise in situations that involve multiple self-interested agents. Often the goal is to choose an outcome or make a decision which reflects the agents' preferences. The challenge, however, is that the agents' preferences are private, and agents may try to manipulate the system so as to ensure an outcome or decision which is desirable for themselves, possibly at the expense of others. In the rest of this section we provide an overview of key game theory and mechanism design concepts used in this paper. A more thorough introduction to game theory and mechanism design can be found elsewhere [5].

3.1.1 Game Theory

The field of game theory studies strategic interactions of self-interested agents. We assume that there is a set of self-interested agents, denoted by I. We let $\theta_i \in \Theta_i$ denote the *type* of agent i which is drawn from some set of possible types Θ_i. The type represents the private information and preferences of the agent. An agent's preferences are over *outcomes* $o \in \mathcal{O}$, where $\mathcal{O}$ is the set of all possible outcomes. We assume that an agent's preferences can be expressed by a utility function $u_i(o, \theta_i)$ which depends on both the outcome, o, and the agent's type, θ_i. Agent i prefers outcome o_1 to o_2 when $u_i(o_1, \theta_i) > u_i(o_2, \theta_i)$.

When agents interact, we say that they are playing *strategies.* A strategy for agent i, $s_i(\theta_i)$, is a plan that describes what actions the agent will take for every decision that the agent might be called upon to make, for each possible piece of information that the agent may have at each time it is called to act. That is, a strategy can be thought as a complete contingency plan for an agent. We let Σ_i denote the set of all possible strategies for agent i, and thus $s_i(\theta_i) \in \Sigma_i$. When it is clear from the context, we will drop the θ_i in order to simplify the notation. We let *strategy profile* $s = (s_1(\theta_1), \ldots, s_I(\theta_I))$ denote the outcome that results when each agent i is playing strategy $s_i(\theta_i)$. As a notational convenience we define

$$s_{-i}(\theta_{-i}) = (s_1(\theta_i), \ldots, s_{i-1}(\theta_{i-1}), s_{i+1}(\theta_{i+1}), \ldots, s_I(\theta_I))$$

and thus $s = (s_i, s_{-i})$. We then interpret $u_i((s_i, s_{-i}), \theta_i)$ to be the utility of agent i with type θ_i when all agents play strategies specified by strategy profile $(s_i(\theta_i), s_{-i}(\theta_{-i}))$.

Since the agents are all self-interested, they will try to choose strategies which maximize their own utility. Since the strategies of other agents also play a role in determining the outcome, the agents must take this into account. The *solution concepts* in game theory determine the outcomes that will arise if all agents are rational and strategic. The most well known solution concept is the *Nash equilibrium.* A Nash equilibrium is a strategy profile in which each agent is following a strategy which maximizes its own utility, given its type and the strategies of the other agents.

Definition 13 (Nash Equilibrium). *A strategy profile $s^* = (s_1^*, \ldots, s_I^*)$ is a Nash equilibrium if no agent has incentive to change its strategy, given that no other agent changes. Formally,*

$$\forall i, u_i(s_i^*, s_{-i}^*, \theta_i) \geq u_i(s_i', s_{-i}^*, \theta_i), \forall s_i'.$$

Although the Nash equilibrium is a fundamental concept in game theory, it does have several weaknesses. First, there may be multiple Nash equilibria and so agents may be uncertain as to which equilibrium they should play. Second, the Nash equilibrium implicitly assumes that agents have perfect information about all other agents, including the other agents' preferences.

A stronger solution concept in game theory is the *dominant-strategy equilibrium.* A strategy s_i is said to be *dominant* if by playing it, the utility of agent i is maximized no matter what strategies the other agents play.

Definition 14 (Dominant Strategy). *Strategy s_i^* is dominant if* $u_i(s_i^*, s_{-i}, \theta_i) \geq u_i(s_i', s_{-i}, \theta_i)$ $\forall s_{-i}$, $\forall s_i'$

A dominant-strategy equilibrium is a strategy profile where each agent is playing a dominant strategy. This is a very robust solution concept since it makes no assumptions about what information the agents have available to them, nor does it assume that all agents know that all other agents are being rational (*i.e.* trying to maximize their own expected utility). However, there are many strategic settings where no agent has a dominant strategy.

A third solution concept is the *Bayes-Nash equilibrium.* In the Bayes-Nash equilibrium the assumption made for the Nash equilibrium, that all agents know the preferences of others, is relaxed. Instead, we assume that there is some common prior $F((\Theta_1, \ldots, \Theta_I))$, such that the agents' types are distributed according to F. Then, in equilibrium, each agent chooses the strategy that maximizes it's expected utility given the strategies other agents are playing and the prior F.

Definition 15 (Bayes-Nash Equilibrium). *A strategy profile $s^* = (s_i^*, s_{-i}^*)$ is a Bayes-Nash equilibrium if $\forall \theta_i, \forall s_i'$:*
$E_{\theta_{-i}}[u_i((s_i^*(\theta_i), s_{-i}^*(\cdot)), \theta_i)] \geq E_{\theta_{-i}}[u_i((s_i'(\theta_i), s_{-i}^*(\cdot)), \theta_i)]$

3.1.2 Mechanism Design

The problem that mechanism design studies is how to ensure that a desirable system-wide outcome or decision is made when there are a group of self-interested agents who have preferences over the outcomes. In particular, we often want the outcome to depend on the preferences of the agents. This is captured by a *social choice function.*

Definition 16 (Social Choice Function). *A* social choice function *is a rule $f : \Theta_1 \times \ldots \times \Theta_I \rightarrow \mathcal{O}$, that selects some outcome $f(\theta) \in \mathcal{O}$, given agent types $\theta = (\theta_1, \ldots, \theta_I)$.*

The challenge, however, is that the types of the agents (the $\theta_i's$) are private and are known only to the agents themselves. Thus, in order to select an outcome with the social choice function, one has to rely on the agents to reveal their types. However, for a given social choice function, an agent may find that it is better off if it does not reveal its type truthfully, since by lying it may be able to cause the social choice function to choose an outcome that it prefers. Instead of trusting the agents to be truthful, we use a *mechanism* to try to reach the correct outcome.

A mechanism $\mathcal{M} = (\Sigma, g(\cdot))$ defines the set of allowable strategies that agents can chose, with $\Sigma = \Sigma_1 \times \cdots \times \Sigma_I$ where Σ_i is the strategy set for agent i, and an outcome function $g(s)$ which specifies an outcome o for each possible strategy profile $s = (s_1, \ldots, s_I) \in \Sigma$. This defines a game in which agent i is free to select any strategy in Σ_i, and, in particular, will try to select a strategy which will lead to an outcome that maximizes its own utility. We say that a mechanism *implements* social choice function f if the outcome induced by the mechanism is the same outcome that the social choice function would have returned if the true types of the agents were known.

Definition 17 (Implementation). *Mechanism $\mathcal{M} =$* $(\Sigma, g(\cdot))$ implements *social choice function f if there exists an equilibrium s^* such that*

$$g(s^*(\theta)) = f(\theta) \; \forall \theta \in \Theta.$$

While the definition of a mechanism puts no restrictions on the strategy spaces of the agents, an important class of mechanisms are the *direct-revelation mechanisms* (or simply *direct mechanisms*).

Definition 18 (Direct-Revelation Mechanism). *A* direct-revelation mechanism *is a mechanism in which $\Sigma_i = \Theta_i$ for all i, and $g(\theta) = f(\theta)$ for all $\theta \in \Theta$.*

In words, a direct mechanism is one where the strategies of the agents are to announce a type, θ_i' to the mechanism. While it is not necessary that $\theta_i' = \theta_i$, the important *Revelation Principle* states that if a social choice function, $f(\cdot)$, can be implemented, then it can be implemented by a direct mechanism where every agent reveals its true type [5]. In such a situation, we say that the social choice function is *incentive compatible.*

Definition 19 (Incentive Compatible). *The social choice function $f(\cdot)$ is* incentive compatible *(or* truthfully implementable*) if the direct mechanism $\mathcal{M} = (\Theta, g(\cdot))$ has an equilibrium s^* such that $s_i^*(\theta_i) = \theta_i$.*

If the equilibrium concept is the dominant-strategy equilibrium, then the social choice function is *strategy-proof.* In this paper we will on occasion call a mechanism incentive-compatible or strategy-proof. This means that the social choice function that the mechanism implements is incentive-compatible or strategy-proof.

3.2 Mechanism Design for Argumentation

In this section we define the mechanism design problem for abstract argumentation. In particular, we specify the agents' type spaces and utility functions, what sort of strategic behavior agents might indulge in, as well as the kinds of social choice functions we are interested in implementing.

We define a mechanism with respect to an argumentation framework $\langle \mathcal{A}, \rightharpoonup \rangle$ with semantics $\mathcal{S}$, and we assume that there is a set of I self-interested agents. We define an agent's type to be its set of arguments.

Definition 20 (Agent Type). *Given an argumentation framework $\langle \mathcal{A}, \rightharpoonup \rangle$, the* type *of agent i, $\mathcal{A}_i \subseteq \mathcal{A}$, is the set of arguments that the agent is capable of putting forward.*

Given the agents' types (argument sets) a social choice function f maps a type profile into a subset of arguments;

$$f : 2^{\mathcal{A}} \times \ldots \times 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$$

While our definition of an argumentation mechanism will allow for generic social choice functions which map type profiles into subsets of arguments, we will be particularly interested in *argument acceptability* social choice functions.

Definition 21 (Argument Acc. SCF). *Given an argumentation framework $\langle \mathcal{A}, \rightharpoonup \rangle$ with semantics $\mathcal{S}$, and given a type profile $(\mathcal{A}_1, \ldots, \mathcal{A}_I)$, the* argument acceptability social choice function *f is defined as the set of acceptable arguments given the semantics $\mathcal{S}$. That is,*

$$f(\mathcal{A}_1, \ldots, \mathcal{A}_I) = Acc(\langle A_1 \cup \ldots \cup \mathcal{A}_I, \rightharpoonup \rangle, \mathcal{S}).$$

As is standard in the mechanism design literature, we assume that agents have preferences over the outcomes $o \in 2^{\mathcal{A}}$, and we represent these preferences using utility functions where $u_i(o, \mathcal{A}_i)$ denotes agent i's utility for outcome o when its type is argument set $\mathcal{A}_i$.

Agents may not have incentive to reveal their true type because they may be able to influence the final argument status assignment by lying, and thus obtain higher utility. There are two ways that an agent can lie in our model. On one hand, an agent might create new arguments that it does not have in its argument set. In the rest of the paper we will assume that there is an *external verifier* that is capable of checking whether it is possible for a particular agent to actually make a particular argument. If an agent is caught making up arguments then it will be removed from the mechanism. For example, in a court of law, any act of perjury by a witness is punished, at the very least, by completely discrediting all evidence produced by the witness. For all intents and purposes this assumption (also made by Glazer and Rubinstein [4]) removes the incentive for an agent to make up facts.

A more insidious form of lying occurs when an agent decides to *hide* some of its arguments. By refusing to reveal certain arguments, an agent might be able to break defeat chains in the argument framework, thus changing the final set of acceptable arguments. For example, a witness may hide evidence that implicates the defendant if the evidence also undermines the witness's own character. This type of lie is almost impossible to detect in practice, and it is this form of strategic behaviour that we will be the most interested in.

As mentioned in the previous subsection, a strategy of an agent specifies a complete plan that describes what action the agent takes for every decision that a player might be called upon to take, for every piece of information that the player might have at each time that it is called upon to act. In our model, the actions available to an agent involve announcing sets of arguments. Thus a strategy, $s_i \in \Sigma_i$ for agent i would specify for each possible subset of arguments that could define its type, what set of arguments to reveal. For example, a strategy might specify that an agent should reveal only half of its arguments without waiting to see what other agents are going to do, while another strategy might specify that an agent should wait and see what arguments are revealed by others, before deciding how to respond. In particular, beyond specifying that agents are not allowed to make up arguments, we place no restrictions on the allowable strategy spaces, when we initially define an argumentation mechanism. Later, when we talk about *direct* argumentation mechanisms we will further restrict the strategy space.

We are now ready to define our argumentation mechanism. We first define a generic mechanism, and then specify a direct argumentation mechanism, which due to the Revelation Principle, is the type of mechanism we will study in the rest of the paper.

Definition 22 (Argumentation Mechanism). *Given an argumentation framework $AF = \langle \mathcal{A}, \rightharpoonup \rangle$ and semantics $\mathcal{S}$, an* argumentation mechanism *is defined as*

$$\mathcal{M}_{AF}^{\mathcal{S}} = (\Sigma_1, \ldots, \Sigma_I, g(\cdot))$$

where Σ_i is an argumentation strategy space of agent i and $g : \Sigma_1 \times \ldots \Sigma_I \rightarrow 2^{\mathcal{A}}$.

Note that in the above definition, the notion of argumentation strategy is broadly construed and would depend on the protocol used. In a *direct* mechanism, however, the strategy spaces of the agents are restricted so that they can only reveal a subset of arguments.

Definition 23 (Dir. Argumentation Mechanism). *Given an argumentation framework $AF = \langle \mathcal{A}, \rightharpoonup \rangle$ and semantics $\mathcal{S}$, an* argumentation mechanism*,*

$$\mathcal{M}_{AF}^{\mathcal{S}} = (\Sigma_1, \ldots, \Sigma_I, g(\cdot))$$

where $\Sigma_i = 2^{\mathcal{A}_i}$ and $g : \Sigma_1 \times \ldots \Sigma_I \rightarrow 2^{\mathcal{A}}$.

In Table 1, we summarise the mapping of multi-agent abstract argumentation as a mechanism design problem.

4. A SCEPTICAL DIRECT ARGUMENTATION MECHANISM

In this section, we specify a direct-revelation argumentation mechanism, in which agents' strategies are to reveal sets

MD Concept	**ArgMD Instantiation**
Agent type $\theta_i \in \Theta_i$	Agent's arguments $\theta_i = \mathcal{A}_i \subseteq \mathcal{A}$
Outcome $o \in \mathcal{O}$	Accepted arguments $Acc(.) \subseteq \mathcal{A}$
Utility $u_i(o, \theta_i)$	Preferences over $2^{\mathcal{A}}$ (what arguments end up being accepted)
Social choice function $f : \Theta_1 \times \ldots \times \Theta_I \to \mathcal{O}$	$f(\mathcal{A}_1, \ldots, \mathcal{A}_I) = Acc(\langle \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_I, \rightharpoonup\rangle, \mathcal{S})$. by some argument acceptability criterion
Mechanism $\mathcal{M} = (\Sigma, g(\cdot))$ where $\Sigma = \Sigma_1 \times \cdots \times \Sigma_I$ and $g \ : \ \Sigma \to \mathcal{O}$	Σ_i is an argumentation strategy, $g \ : \ \Sigma \to 2^{\mathcal{A}}$
Direct mechanism: $\Sigma_i = \Theta_i$	$\Sigma_i = 2^{\mathcal{A}}$ (every agent reveals a set of arguments)
Truth revelation	Revealing $\mathcal{A}_i$

Table 1: Abstract argumentation as a mechanism

of arguments, and where the mechanism calculates the outcome using sceptical (grounded) semantics. We show that, in general, this mechanism gives rise to strategic manipulation. We then prove that under certain conditions, this mechanism is strategy proof.

As we stated earlier, an agent's type $\mathcal{A}_i$ determines its preferences over outcomes via a utility function $u_i(o, \mathcal{A}_i)$. The utility may be defined in a variety of ways. In this paper, we only consider one type of preference, in which i attempts to maximise the number of accepted arguments from $\mathcal{A}_i$. This criterion holds, for example, in cases where agents lose credibility when they present arguments that end up being rejected. Thus an agent must weigh the potential benefit of presenting an argument against the "loss of face" resulting from potential rejection. Debates in political campaigns exhibit this kind of characteristics.

Definition 24 (Acceptability maximising prefs.). *An agent i has* individual acceptability maximising preferences *if and only if* $\forall o_1, o_2 \in \mathcal{O}$ *such that* $|o_1 \cap \mathcal{A}_i| \geq |o_2 \cap \mathcal{A}_i|$, *we have* $u_i(o_1, \mathcal{A}_i) \geq u_i(o_2, \mathcal{A}_i)$.

In a direct argumentation mechanism, each agent i's available actions are $\Sigma_i = 2^{\mathcal{A}_i}$. We will refer to a specific action (*i.e.* set of declared arguments) as $\mathcal{A}_i^\circ \in \Sigma_i$.

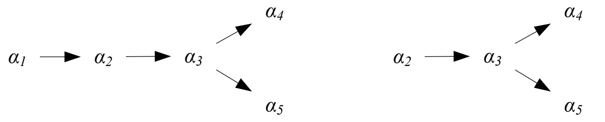
We now present a direct mechanism for argumentation based on the sceptical argument evaluation criteria. The mechanism then calculates the set of sceptically acceptable arguments (*i.e.* it calculates the grounded extension) given the union of all arguments revealed by agents.

Definition 25 (Sceptical Direct Arg. Mech.). *A* sceptical direct argumentation mechanism *for argumentation framework* $\langle \mathcal{A}, \rightharpoonup \rangle$ *is* $\mathcal{M}_{AF}^{grnd} = (\Sigma_1, \ldots, \Sigma_I, g(.))$ *where:*

- $\Sigma_i \in 2^{\mathcal{A}}$ *is the set of strategies available to each agent;*
- $g \ : \ \Sigma_1 \times \cdots \times \Sigma_I \to 2^{\mathcal{A}}$ *is an outcome rule defined as:* $g(\mathcal{A}_1^\circ, \ldots, \mathcal{A}_I^\circ) = Acc(\langle \mathcal{A}_1^\circ \cup \cdots \cup \mathcal{A}_I^\circ, \rightharpoonup\rangle, \mathcal{S}^{grnd})$ *where* $\mathcal{S}^{grnd}$ *denotes sceptical acceptability semantics.*

Let us now consider aspects of incentives using mechanism $\mathcal{M}_{AF}^{grnd}$ through an example.

Example 26. *Consider a sceptical direct argumentation mechanism with three agents x, y and z with types $\mathcal{A}_x = \{\alpha_1, \alpha_4, \alpha_5\}$, $\mathcal{A}_y = \{\alpha_2\}$ and $\mathcal{A}_z = \{\alpha_3\}$ respectively. And suppose that the defeat relation is defined as follows: $\rightharpoonup = \{(\alpha_1, \alpha_2), (\alpha_2, \alpha_3), (\alpha_3, \alpha_4), (\alpha_3, \alpha_5)\}$. If each agent reveals its true type (i.e. $\mathcal{A}_x^\circ = \mathcal{A}_x$; $\mathcal{A}_y^\circ = \mathcal{A}_y$; and $\mathcal{A}_z^\circ = \mathcal{A}_z$), then we get the argument graph depicted in Figure 3(a). The mechanism outcome rule produces the outcome $o = \{\alpha_1, \alpha_3\}$.*



(a) Full revelation (b) Argument α_1 withheld

Figure 3: Hiding an argument is beneficial

If agents have individual acceptability maximising preferences, with utilities equal to the number of arguments accepted, then: $u_x(o, \{\alpha_1, \alpha_4, \alpha_5\}) = 1$; $u_y(o, \{\alpha_3\}) = 1$; and $u_z(o, \{\alpha_2\}) = 0$.

It turns out that the mechanism is susceptible to strategic manipulation, even if we suppose that agents do not lie by making up arguments (*i.e.* they may only withhold some arguments). In this case, for both agents y and z, revealing their true types weakly dominates revealing nothing at all. However, it turns out that agent x is better off revealing $\{\alpha_4, \alpha_5\}$. By withholding α_1, the resulting argument network becomes as depicted in Figure 3(b), for which the output rule produces the outcome $o' = \{\alpha_2, \alpha_4, \alpha_5\}$. This outcome yields utility 2 to agent x, which is better than the truth-revealing strategy.

Remark 27. *Mechanism $\mathcal{M}_{AF}^{grnd}$ is not strategy-proof.*

An interesting question, therefore, is whether mechanism $\mathcal{M}_{AF}^{grnd}$ has a truth-revealing property (incentive compatibility, or strategy-proofness) under some additional condition. As we shall demonstrate below, it turns out that there is a reasonable condition, and that it fully characterises strategy-proof sceptical argumentation mechanisms. Before we present our main result, we first need to present a few definitions and lemmas.

Definition 28 (Indirect defeat and defence [3]). *Let $\alpha, \beta \in \mathcal{A}$. We say that α* indirectly defeats *β, written $\alpha \hookrightarrow \beta$, if and only if there is an odd-length path from α to β in the argument graph. We say that α* indirectly defends *β if and only if there is an even-length path (with non-zero length) from α to β in the argument graph.*

Definition 29 (Parents & initial arguments [1]). *Given an argumentation framework $AF = \langle \mathcal{A}, \rightharpoonup \rangle$ and an argument $\alpha \in \mathcal{A}$, the* parents *of argument α are denoted by $\mathtt{par}_{AF}(\alpha) = \{\beta \in \mathcal{A} \mid \beta \hookrightarrow \alpha\}$. Arguments in AF that have no parents are called* initial *arguments, and are denoted by the set $IN(AF) = \{\alpha \in \mathcal{A} \mid \mathtt{par}_{AF}(\alpha) = \emptyset\}$.*

The following lemma, which is necessary for our subsequent proofs, shows that each acceptable argument is indirectly defended by some initial argument. The lemma states that any acceptable argument is indirectly defended, against each defeater (*i.e.* parent), by some initial argument. This highlights that initial arguments play an important role in the defence of every other acceptable argument.

LEMMA 30. *Let $AF = \langle \mathcal{A}, \rightharpoonup \rangle$ be an argumentation framework. If argument $\alpha \in Acc(AF, \mathcal{S}^{grnd})$ then $\forall P \in \mathtt{par}_{AF}(\alpha)$, $\exists \beta \in IN(AF)$ such that $\beta \hookrightarrow P$.*

PROOF. *Omitted due to space limitations.* □

We now explore what happens when we add a new argument (and its associated defeats) to a given argumentation framework, producing a new argumentation framework. In particular, we are interested in conditions under which arguments acceptable in the first framework are also accepted in the second. We show that this is true under the condition that the new argument does not indirectly defeat arguments acceptable in the first framework. This is stated in the following lemma, the proof of which makes use of Lemma 30.

LEMMA 31. *Let $AF_1 = \langle \mathcal{A}, \rightharpoonup_1 \rangle$ and $AF_2 = \langle \mathcal{A} \cup \{\alpha'\}, \rightharpoonup_2 \rangle$ such that $\rightharpoonup_1 \subseteq \rightharpoonup_2$ and $(\rightharpoonup_2 \setminus \rightharpoonup_1) \subseteq (\{\alpha'\} \times \mathcal{A}) \cup (\mathcal{A} \times \{\alpha'\})$. If α is in the grounded extension of AF_1 and α' does not indirectly defeat α, then α is also in the grounded extension of AF_2.*

PROOF. *Omitted due to space limitations.* □

We are now ready to prove the main result, which states conditions under which $\mathcal{M}_{AF}^{grnd}$ is strategy proof.

THEOREM 32. *Suppose agents have individual acceptability maximising preferences. If each agent's type corresponds to a conflict-free set of arguments which does not include indirect/direct defeats (formally $\forall i \nexists \alpha_1, \alpha_2 \in \mathcal{A}_i$ such that $\alpha_1 \hookrightarrow \alpha_2$), then $\mathcal{M}_{AF}^{grnd}$ is strategy-proof.*

PROOF. *Let $\mathcal{A}'_{-i} = (\mathcal{A}'_1, \ldots, \mathcal{A}'_{i-1}, \mathcal{A}'_{i+1}, \ldots, \mathcal{A}'_I)$ be arbitrary revelations from all agents not including i. We will show that agent i is always best off revealing $\mathcal{A}_i$. That is, no matter what sets of arguments the other agents reveal, agent i is best off revealing its full set of arguments. Formally, we will show that $\forall i \in I$ $u_i(Acc(\langle \mathcal{A}'_1 \cup \cdots \cup \mathcal{A}_i \cup \cdots \cup \mathcal{A}'_I, \rightharpoonup \rangle, \mathcal{S}^{grnd}), \mathcal{A}_i) \geq u_i(Acc(\langle \mathcal{A}'_1 \cup \cdots \cup \hat{\mathcal{A}}_i \cup \cdots \cup \mathcal{A}'_I, \rightharpoonup \rangle, \mathcal{S}^{grnd}), \mathcal{A}_i)$ for any $\hat{\mathcal{A}}_i \subset \mathcal{A}_i$.*

If $\mathcal{A}_i = \emptyset$, then trivially the agent has nothing to reveal. Otherwise, we use induction over the sets of arguments agent i may reveal, starting from an arbitrary single argument. We show that, considering any strategy $\mathcal{A}''_i \subseteq \mathcal{A}_i$, revealing one more argument can only increase i's set of acceptable arguments, i.e. it (weakly) improves i' utility.

- Base Step: *If $\mathcal{A}_i = \{\alpha\}$ for some arbitrary single argument α, then revealing $\mathcal{A}_i$ weakly dominates revealing $\emptyset$. This is because if i does not reveal α, it receives the worst utility value of zero, while revealing α may result in utility 1 if α is accepted.*

- Induction Step: *Suppose that revealing argument set $\mathcal{A}''_i \subseteq \mathcal{A}_i$ weakly dominates revealing any subset of $\mathcal{A}''_i$. We need to prove that revealing any additional argument can increase, but never decrease the agent's utility. In other words, we need to prove that revealing any set $\mathcal{A}'_i$, where $\mathcal{A}''_i \subset \mathcal{A}'_i \subseteq \mathcal{A}_i$ and $|\mathcal{A}'_i| = |\mathcal{A}''_i| + 1$, weakly dominates revealing $\mathcal{A}''_i$.*

  Let α' where $\{\alpha'\} = \mathcal{A}'_i - \mathcal{A}''_i$ be the new argument.

  Let $\alpha'' \in \mathcal{A}''_i \cap Acc(\langle \mathcal{A}'_1 \cup \cdots \cup \mathcal{A}''_i \cup \cdots \cup \mathcal{A}'_I, \rightharpoonup \rangle, \mathcal{S}^{grnd})$ be an arbitrary argument from $\mathcal{A}''_i$ that is found to be sceptically accepted when revealing $\mathcal{A}''_i$. We need to show that after adding α', argument α'' remains sceptically accepted. Formally, we need to show that $\alpha'' \in \mathcal{A}'_i \cap Acc(\langle \mathcal{A}'_1 \cup \cdots \cup \mathcal{A}'_i \cup \cdots \cup \mathcal{A}'_I, \rightharpoonup \rangle, \mathcal{S}^{grnd})$. This is true from Lemma 31, and from the fact that $\mathcal{A}_i$ does not include indirect defeats.

Thus, by induction, revealing the full set $\mathcal{A}_i$ weakly dominates revealing any sub-set thereof. □

Note that in the theorem, $\hookrightarrow$ is over all arguments in $\mathcal{A}$. Intuitively, the condition in the theorem states that each agent's arguments must be consistent, both explicitly and implicitly. Explicit consistency implies that no argument defeats another. Implicit consistency implies that no other agent can present an argument that reveals an indirect defeat among one's own arguments (in more concrete settings, this may be interpreted as revealing a fallacy in one's arguments).

We now prove the converse of the above theorem. That is, we show that if a mechanism that implements the sceptical social choice function is strategy proof, then it must satisfy the condition that individual agents do not have arguments that indirectly defeat one another.

THEOREM 33. *Let I be a set of agents with individual maximising preferences. Let $\mathcal{M}_{AF}^{grnd}$ be a mechanism that implements the sceptical social choice function f. If $\mathcal{M}_{AF}^{grnd}$ is strategy-proof, then no agent type includes indirectly self-defeating arguments.*

PROOF. *We will prove this by contradiction. We are given that $\mathcal{M}_{AF}^{grnd}$ is a strategy-proof mechanism that implements the sceptical social choice function f. Assume that agents have types which include indirectly self-defeating arguments. In particular, consider the argument graph shown in Figure 3 and assume that there are three agents such that: $\mathcal{A}_1 = \{\alpha_1, \alpha_4, \alpha_5\}$, $\mathcal{A}_2 = \{\alpha_2\}$, $\mathcal{A}_3 = \{\alpha_3\}$.*

Since mechanism $\mathcal{M}_{AF}^{grnd}$ is strategy-proof then for all $\mathcal{A}_i$

$$u_i(g(\mathcal{A}_i, \mathcal{A}_{-i}), \mathcal{A}_i) \geq u_i(g(\mathcal{A}'_i, \mathcal{A}_{-i}), \mathcal{A}_i)$$

for all i, for all $\mathcal{A}'_i \neq \mathcal{A}_i$ and for all $\mathcal{A}_{-i}$. Thus, the following constraint must hold:
$u_1(g(\{\alpha_1, \alpha_4, \alpha_5\}, \mathcal{A}_2, \mathcal{A}_3), \{\alpha_1, \alpha_4, \alpha_5\}) \geq$
$u_1(g(\{\alpha_4, \alpha_5\}, \mathcal{A}_2, \mathcal{A}_3), \{\alpha_1, \alpha_4, \alpha_5\})$.

Since agents have individual maximising preferences, this means that $|\mathcal{A}_1 \cap g(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)| \geq |\mathcal{A}_1 \cap g(\{\alpha_4, \alpha_5\}, \mathcal{A}_2, \mathcal{A}_3)|$.

However, for this constraint to hold, it must be the case that $g(\cdot) \neq f(\cdot)$ where f is the sceptical social choice function, since

$$\begin{aligned} f(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3) &= \{\alpha_1, \alpha_3\} \\ f(\{\alpha_4, \alpha_5\}, \mathcal{A}_2, \mathcal{A}_3) &= \{\alpha_2, \alpha_4, \alpha_5\} \end{aligned}$$

and so $|\mathcal{A}_1 \cap f(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)| < |\mathcal{A}_1 \cap f(\{\alpha_4, \alpha_5\}, \mathcal{A}_2, \mathcal{A}_3)|$.

This implies that the mechanism $\mathcal{M}_{AF}^{grnd}$ does not implement the sceptical social choice function, which is a contradiction. Therefore, no agent type can include indirectly self-defeating arguments. □

From Theorems 32 and 33 above, we see that we have the full characterisation of strategy-proof mechanisms for sceptical argumentation frameworks in which agents have individual acceptability maximising preferences. This is formalised in the following corollary.

Corollary 34. *Let I be a set of agents with individual maximising preferences. Let $\mathcal{M}_{AF}^{grnd}$ be a mechanism that implements the sceptical social choice function f. $\mathcal{M}_{AF}^{grnd}$ is strategy-proof if and only if none of the argument sets of the agents in I contain indirectly self-defeating arguments.*

Thus, we have presented a sceptical direct argumentation mechanism. We showed that this mechanism is strategy-proof under the condition that individual agents do not have arguments that indirectly defeat one another. This means that the mechanism has an important property: it *maximises the use of the information* available. This property is particularly important in settings where a decision-maker wants to make the most informed decision possible.

5. RELATED WORK

Analysis of strategies in argumentation is scarce, and much of it addresses *heuristic* strategies (*e.g.* assertion attitudes [6]). When compared with this approach (where only a handful of heuristic strategies are analysed), game-theoretic analysis is more thorough. Not only does it take into account the full spectrum of possible agent behaviour, including heuristic strategies, it also provides us with the tools, via mechanism design, to begin designing argumentation frameworks that guarantee certain desirable properties.

To our knowledge, the only other game-theoretic analysis of Dung-style argumentation is by Procaccia and Rosenschein [7]. The authors are interested in the *dynamics* of argument, while we are interested in its *outcomes*. They map Dung's frameworks into extensive-form games of perfect information, and present algorithms for determining equilibria for the argumentation games that are guaranteed to terminate. While the motivation and goals behind their approach is quite different from ours, we believe that it provides a complement to our mechanism-design framework.

In economics, Glazer and Rubinstein [4] explored the mechanism design problem of constructing rules of debate that maximise the probability that a listener reaches the right conclusion given arguments presented by two debaters. They discuss a very restricted setting with 5 possible arguments (with no explicit relationship between them) and 2 outcomes. Most related to our work is the *simultaneous debate* in which the two debaters simultaneously reveal one argument each, with arbitrary rules deciding the outcome. Our approach is more general as it enables simultaneous revelation of an arbitrary number of arguments by an arbitrary number of agents. Moreover, our sceptical mechanism provides a more natural criterion for argument evaluation that exploits the explicit defeat relation among arguments.

It is worthwhile referring to recent work on merging multiple Dung-style argumentation graphs presented by multiple agents [2]. The authors use a combination of graph expansion, distance calculation and voting in order to arrive at a single argumentation framework. The key difference between this work and ours is that agents in the former are cooperative: they do not have conflicting preferences over what the final framework should look like. As such, the possibility of hiding arguments is not discussed.

6. DISCUSSION AND CONCLUSION

We introduced *argumentation mechanism design* (ArgMD), which enables the design and analysis of argumentation mechanisms for self-interested agents. We did this by casting the standard abstract argumentation framework as a (game-theoretic) mechanism. We then defined a particular class of argumentation mechanisms, namely direct argumentation mechanisms. In these mechanisms, agents must decide which arguments to reveal simultaneously, and the mechanism calculates the set of accepted arguments based on some argument acceptability criterion. We then studied a particular *sceptical* direct argumentation mechanism, under a certain class of agent preferences (namely, agents that want to maximise the number of their own accepted arguments). We showed that, in general, agents may have incentive to lie by hiding arguments in an attempt to influence the outcome. We then demonstrated that, under meaningful topological restrictions on the argument graph, the sceptical mechanism becomes *strategy-proof.* We proved that these restrictions are necessary and sufficient to ensure strategy-proofness.

The work presented in this paper is just the beginning of what we envisage to be a growing area at the intersection of game-theory and formal argumentation theory. For the first time in the literature on argumentation frameworks, we can now take the study of strategies seriously when designing argument acceptability rules (or semantics). Without this game-theoretic perspective, using argumentation in real (open) agent systems has been a far-fetched prospect. We envisage that our work, with its new approach to designing argumentation rules, will help bridge the gap between theory and application.

Acknowledgments

The authors are grateful to Pietro Baroni, Andrew Clausen and Sherief Abdallah for useful discussions.

7. REFERENCES

[1] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10–15):675–700, 2007.

[2] S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, and P. Marquis. On the merging of Dung's argumentation systems. *Artificial Intelligence*, 171(10–15):730–753, 2007.

[3] 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.

[4] J. Glazer and A. Rubinstein. Debates and decisions: On a rationale of argumentation rules. *Games and Economic Behavior*, 36:158–173, 2001.

[5] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory.* Oxford University Press, New York NY, USA, 1995.

[6] S. Parsons, M. J. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[7] A. D. Procaccia and J. S. Rosenschein. Extensive-form argumentation games. In *Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS-05), Brussels, Belgium*, pages 312–322, 2005.