Extensive-Form Argumentation Games

Ariel D. Procaccia Jeffrey S. Rosenschein

School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel

Abstract

Two prevalent approaches to automated negotiation are the application of game-theoretic notions and the argumentation-based angle; these two schemes are frequently at odds. An elegant view of argumentation is Dung's abstract argumentation theory [2], which cold-shoulders the internal structure of arguments in favor of the entire debate's global structure. Dung's theory is elaborated by work in dialectical argumentation theory, which focuses on dialogues between two players.

In this paper, we enhance the abovementioned frameworks by considering two-agent settings where each of the agents is identified with a set of arguments. A binary attack relation between arguments is given, as well as (most importantly) a payoff function that assigns real values to every possible valid dialogue. Such *game-based argumentation* frameworks can be lucidly realized as games in extensive form (marrying, in a sense, the game-theoretic and argumentation-based approaches). We investigate specialized notions of "maximally-defendable" sets of arguments, and the correlation between the properties of the agents' argument sets and the size of the associated game tree. Furthermore, algorithmic issues are considered: we present simplifications and efficient solutions for our argumentation games, and discuss a special case where the representation of the framework is logarithmic in the size of the associated game tree.

1 Introduction

Negotiation between agents has emerged in recent years as a method of resolving scenarios where agents have conflicting interests but may benefit by cooperating. The work on automated negotiation can be classified into three general approaches [13]: game-theoretic [10, 14, 15], heuristic-based [4, 5, 9], and argumentation-based [11, 12, 16]. A major advantage of argumentation-based negotiation over other styles of negotiation is that positions can be justified, as well as changed during the negotiation [19].

Argumentation-based negotiation itself is a field that encompasses many techniques and a large amount of research [13]. One well-known argumentation framework is the abstract argumentation framework, originally introduced by Dung [2]. This approach shifts the focus from the internal structure of individual arguments to the structure of the entire argument.[1] Informally, an abstract argumentation framework consists of a set of arguments, and an attack relation between arguments. The question asked with regard to specific arguments is whether they are "acceptable", and with regard to sets of arguments — whether they are "defendable".

Dung's abstract approach is static and monological in nature. Therefore, a natural step forward is the study of dialectical argumentation, which revolves around dialogues between two players. This approach was formalized in [8], but early related work can also be found in [18]; more recent noteworthy papers include [3, 1]. These papers present dialogues as sequences of arguments, which satisfy restrictions imposed by a legal-move function. A dialogue is won by one of the players if he was the last to make a move, and there is no legal response to this argument.

'Logic games' is another related approach that has received much attention [7]. Although work in that area is perhaps more game-theoretic than the abovementioned papers, it views arguments as logical entities — an approach that diverges from the abstract path we wish to take here.

[1] We use the term 'argument' ambiguously: it might mean a specific assertion or an entire debate.

In this paper, we consider argumentation frameworks that consist of two different sets of arguments (instead of the traditional single set), each being the set of arguments available to one of two agents. As in dialectical argumentation, at each stage in the dialogue one of the agents advances an argument that counters the last argument produced by its rival.

We abandon the setting where agents either win or lose arguments. In our view, a player's reward for a specific dialogue can be any real number. An argument resulting in a payoff of $(0.7, 0.3)$, for example, could mean that agent 1 is 70% "right", while agent 2 is only 30% "right" (or 70% "wrong"). Such flexible payoffs may be especially useful when the agents are negotiating over a divisible good.

The argumentation game, as we see it, can be described in a clear-cut manner as a game in extensive form, where the actions available at a node owned by one of the agents are a subset of that agent's argument set. These ideas are the basis of our *extensive-form argumentation games* approach, which in a sense combines the game-theoretic and argumentation-based styles of negotiation. The major advantage in representing the argument as a game in extensive form is that it enables the utilization of existing methods from algorithmic game theory in order to solve the associated game. For instance, in the perfect information setting, which is the focus of this paper, we can derive, using backward induction, a Nash equilibrium in pure strategies.

In practical settings it is only reasonable to desire that the argument be finite. We approach the question of whether the argumentation game is infinite from two directions: the properties of the agents' argument sets, and the relations between these two sets. By defining 'local' versions of Dung's semantics, we find that when the agents' argument sets satisfy some notions of defendability, the resulting argument is infinite. We characterize the relations between the two argument sets by introducing the "interaction graph", and show that the game is infinite if and only if the graph is cyclic.

We also explore the algorithmic aspects that arise from our setup. Several methods to simplify infinite argumentation games are suggested. Another issue that is tackled is the resolution of game-based arguments in time proportional to the size of the representation of the argumentation framework, instead of the size of the associated game tree. We discuss a problematic case where the payoff function can be represented compactly, but provide a dynamic programming algorithm that resolves it efficiently.

The paper proceeds as follows. In Section 2, we give an overview of Dung's abstract argumentation theory. In Section 3, we introduce our game-based argumentation frameworks and present our results. In Section 4, we suggest two major directions for future research.

2 Preliminaries

In this section we review some of the concepts of Dung's theory[2] of abstract argumentation [2]. The theory is concerned with argumentation frameworks governed by an attack relation between arguments; it is not concerned with the internal structure of arguments. This theory is the starting point of our construction.

Definition 1. An *argumentation framework* $\mathcal{A}$ is a pair $\langle AR, \rightarrow \rangle$ where AR is a set of arguments, and $\rightarrow$ is a binary relation on AR. The expression $a \rightarrow b$ is pronounced "a attacks b" or "b is attacked by a".

Informally, a set of arguments is *acceptable* if it can be defended against attacks. Abstract argumentation theory is concerned with maximally-defendable sets of arguments, but there are several possible semantics, each inducing different such sets.

Definition 2.

1. An argument $a \in AR$ is *attacked by a set of arguments* S (denoted $S \rightarrow a$) iff S contains an attacker of a.

2. An argument $a \in AR$ is *acceptable* with respect to a set S of arguments iff for all $b \in AR$: if $b \rightarrow a$ then $S \rightarrow b$.

[2] We try to adhere to Dung's original formulation, although some of the definitions have slightly evolved in recent papers.

3. A set S of arguments is said to be *conflict-free* iff there are no arguments a and b in S such that $a \to b$.

4. A conflict-free set of arguments S is *admissible* iff each argument in S is acceptable with respect to S.

5. A *preferred extension* of an argumentation framework $\mathcal{A}$ is a $\subseteq$-maximal admissible set of $\mathcal{A}$.

6. A conflict free set of arguments S is called a *stable extension* iff S attacks each argument which does not belong to S.

7. An admissible set of arguments S is called a *complete extension* iff for every argument $a \in AR$: if a is acceptable with respect to S then $a \in S$.

The following result, which appeared in [2], will be relevant to this study.

Proposition 3. *Every preferred extension is a complete extension, but not vice-versa.*

3 Results

3.1 Game-Based Argumentation Frameworks

The basis of our contribution is the augmentation of the abstract argumentation framework with additional constructs, which take into consideration some of the intricacies of interaction between two agents. The definitions can easily be extended to deal with more than two agents, but that is beyond the scope of this paper.

The two agents in our set of agents $N = \{1, 2\}$ are each endowed with an argument set; the sets associated with agents 1 and 2 are denoted by AR_1 and AR_2, respectively. The interaction between the agents is represented as a sequence of arguments: agent 1 begins by presenting an argument $a_1 \in AR_1$, agent 2 counters with an argument $a_2 \in AR_2$ which attacks a_1, agent 1 retaliates with an argument $a_3 \in AR_1$ such that $a_3 \to a_2$, and so forth. The formalization of this concept is inspired by the existing definition of *dialogue history* [19], and relies on the existence of the sets AR_1 and AR_2, in addition to an attack relation between arguments.

Definition 4. A sequence of arguments $(a_1, a_2, \ldots, a_l)$ is a *dialogue* iff $a_i \in AR_1$ when i is odd, $a_i \in AR_2$ when i is even, and for all $i = 2, 3, \ldots, l$: $a_i \to a_{i-1}$.

Remark 5. We say a dialogue $D = (a_1, a_2, \ldots, a_l)$ *begins* with a_1 and *ends* with a_l, and that $|D| = l$.

In our argumentation setting, either agent can terminate the dialogue at any stage. This assumption can be incorporated by assuming the existence of two distinguished arguments $t_1 \in AR_1$ and $t_2 \in AR_2$, such that for all $a \in AR_i$, $t_{3-i} \to a$.

In many argument settings, the outcome of the dialogue depends on the entire history of the arguments employed by the participants. We model this by associating an argumentation setting with a utility function U, which assigns payoffs to both agents for each possible terminated dialogue — any dialogue that ends either with t_1 or t_2. These payoffs are not necessarily "win" and "lose", but may express different degrees of success in the argument.

The following definition unifies the ideas that were presented above.

Definition 6. A *game-based argumentation (GBA) framework* $\mathcal{A}$ is a tuple

$$\langle AR, \to, AR_1, AR_2, U \rangle,$$

where:

1. $\langle AR, \to \rangle$ is an argumentation framework.

2. For $i = 1, 2$: $AR_i \subset AR$, and AR_i is finite. Furthermore, there exist $t_1 \in AR_1$ and $t_2 \in AR_2$ such that t_1 attacks all arguments in AR_2 but is attacked by none, and t_2 attacks all arguments in AR_1 but is attacked by none.

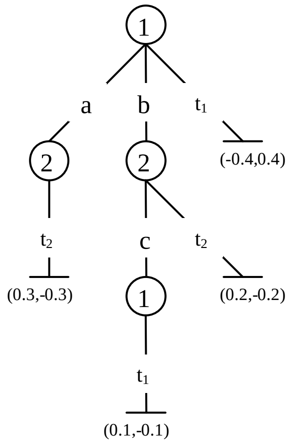


Figure 1: An illustration of the mapping from GBA frameworks to games in extensive form. The parameters of the framework $\mathcal{A}$ are: $AR = \{a, b, c\}$, $AR_1 = \{a, b\}$, $AR_2 = \{c\}$, $\rightarrow = \{(c, b)\}$; the values of U are as follows: $(t_1) \mapsto (-0.4, 0.4)$, $(a, t_2) \mapsto (0.3, -0.3)$, $(b, t_2) \mapsto (0.2, -0.2)$, $(b, c, t_1) \mapsto (0.1, -0.1)$. The figure shows the associated game tree $T_{\mathcal{A}}$.

3. U is a mapping from the set of all valid dialogues ending with t_1 or t_2 to

$$\{(x_1, x_2) \in \mathbb{R}^2 : \ x_1 \in [-0.5, 0.5] \ \wedge \ x_1 + x_2 = 0\}.$$

Remark 7. The introduction of the distinguished arguments t_1 and t_2 into the discussion is a mathematical formality. Below, their existence will be ignored when discussing specific GBA frameworks. These arguments merely symbolize an option to terminate the argument — an option that is always available to both agents.

Remark 8. We require that AR_1 and AR_2 both be finite. Essentially, we want each agent to have a finite number of possible actions at each turn. It is not sufficient to simply demand that each argument be attacked by a finite number of arguments, since agent 1 could, in that situation, still have an infinite number of moves on the first turn.

Remark 9. Conceptually, the payoffs of the agents are both nonnegative and sum to one. If an agent is totally convincing, its payoff would be 1 and its counterpart's 0. Different degrees of success would be represented by different payoffs; for example, a payoff of 0.7 to agent 1 and 0.3 to agent 2 is possible. We chose to represent the setting as a zero-sum game; this has computational implications that will be made apparent later. A payoff of (x_1, x_2) is logically equivalent to $(0.5 + x_1, 0.5 + x_2)$. Thus, a payoff of $(0.2, -0.2)$ is equivalent to $(0.7, 0.3)$.

The function U will sometimes be interpreted as a pair of functions U^1 and U^2, whose values are the payoffs of agents 1 and 2, respectively.

A very natural way to analyze the interaction between the two agents in a GBA framework is by a game in extensive form. Let $\mathcal{A} = \langle AR, \rightarrow, AR_1, AR_2, U \rangle$ be an argumentation framework; we describe how to construct the associated game tree, $T_{\mathcal{A}}$. Every decision node of even depth in the tree is owned by agent 1, and every decision node of odd depth is owned by agent 2. In particular, the root is owned by agent 1. The possible actions available to agent 1 at the root are all the arguments in AR_1; these arguments are associated with the outgoing edges incident on the root. For every other decision node owned by agent $i \in N$, the possible moves are all arguments in AR_i which attack the very last argument in the dialogue — the one associated with the incoming edge incident on the current decision node. Not surprisingly, the sequence of edges on a path from the root to a terminal node is associated with a dialogue. The parent of a terminal node is either t_1 or t_2, and the payoff for a terminal node is the value of U on the dialogue induced by the path from the root to the terminal node. See Figure 1 for an illustration.

Remark 10. The mapping between GBA frameworks and game trees is clearly not surjective, but it is not injective either. To see that it is not injective, consider two different GBA frameworks $\mathcal{A}_1$

and $\mathcal{A}_2$, both with $AR = \{a, b, c\}$, $AR_1 = \{a\}$ and $AR_2 = \{b, c\}$. $\mathcal{A}_1$ has $\to = \{(a, b), (b, a)\}$, and $\mathcal{A}_2$ has $\to = \{(a, b), (b, a), (a, c)\}$. It holds that $T_{\mathcal{A}_1} \equiv T_{\mathcal{A}_2}$.

In this discussion, we sometimes consider the agents' two argument sets to be disjoint and nonempty; we find this to be somewhat intuitive in many real-life settings.

Definition 11. A GBA framework is *normal* if AR_1 and AR_2 are disjoint and nonempty.

3.2 Local Semantics and the Structure of the Game Tree

In game-based argumentation frameworks, we are often interested only in the feud between the two opposing argument sets AR_1 and AR_2, and are somewhat indifferent to their attack and defeat relations with other arguments in $AR \setminus (AR_1 \cup AR_2)$. This motivates us to consider "local" versions of the semantics reviewed in Section 2.

Definition 12. Let $\mathcal{A} = \langle AR, \to, AR_1, AR_2, U \rangle$ be a GBA framework.

1. An argument $a \in AR$ is *locally-acceptable* with respect to a set of arguments S iff for each argument $b \in AR_1 \cup AR_2$: if b attacks a then b is attacked by S.
2. A conflict-free set of arguments S is *locally-admissible* iff each argument in S is locally-acceptable with respect to S.
3. A *locally-preferred extension* of $\mathcal{A}$ is a $\subseteq$-maximal locally-admissible set of $\mathcal{A}$.
4. A conflict-free set of arguments S is called a *stable extension* iff S attacks each argument in $AR_1 \cup AR_2$ which does not belong to S.
5. A locally-admissible set S of arguments is called a *locally-complete extension* iff each argument, which is locally-acceptable with respect to S, belongs to S.

Proposition 13.

1. *An admissible set is locally-admissible, and a stable set is locally-stable.*
2. *Let A be a preferred extension. Then there is some locally-preferred extension B such that $A \subseteq B$.*
3. *The notions of complete and locally-complete extensions are incomparable: there are complete extensions that are not locally-complete, and vice-versa. Moreover, this is true even for normal GBA frameworks.*

Proof.

1. Obvious.
2. A is admissible, therefore it is locally-admissible; but a locally-preferred extension is a maximal locally-admissible set.
3. In the following normal GBA framework, AR_1 is complete[3] but not locally-complete: $AR = \{a, b, c, d, e\}$, $AR_1 = \{a\}$, $AR_2 = \{b\}$, and

$$\to = \{(a, b), (c, d), (d, e), (e, c)\}.$$

   On the other hand, AR_1 in the following GBA framework is locally-complete, but it is not complete: $AR = \{a, b, c, d\}$, $AR_1 = \{a\}$, $AR_2 = \{b, c\}$, and

$$\to = \{(b, c), (c, b), (b, d), (d, a)\}.$$

   See Figure 2 for an illustration.

□

[3]In fact, AR_1 in this example is a preferred extension, which by Theorem 3 is a stronger notion than completeness.

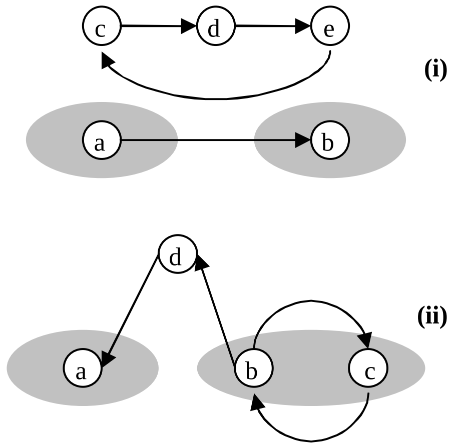


Figure 2: **(i)** $AR_1 = \{a\}$, $AR_2 = \{b\}$; AR_1 is complete but not locally complete. **(ii)** $AR_1 = \{a\}$, $AR_2 = \{b, c\}$; AR_1 is locally-complete but not complete.

Although in theory our agents could bicker and debate for all eternity, in practice it is desirable that negotiations terminate in finite time. Therefore, we would like our dialogues — and our game trees — to be finite.

Definition 14. A GBA framework $\mathcal{A}$ is *finite* iff $T_{\mathcal{A}}$ is finite. A framework is *infinite* if it is not finite.

As we shall shortly prove, an argumentation framework is often infinite. Moreover, the game tree sometimes satisfies an even stronger notion of infinity:

Definition 15. A GBA framework is *subgame-infinite* if for any decision node v in $T_{\mathcal{A}}$, the subtree of $T_{\mathcal{A}}$ rooted in v is infinite.

Proposition 16. *Let $\mathcal{A}$ be a normal GBA framework such that AR_1 and AR_2 are both locally-stable extensions. Then $\mathcal{A}$ is subgame-infinite.*

Proof. Let v be a decision node, and suppose first that v is not the root. To prove that there is an infinite path that begins with v, it is sufficient to prove that any decision node v' which is not the root has at least one child which is a decision node. Indeed, assume w.l.o.g. that v' is owned by agent 2. The incoming edge incident on v' is labeled by $a \in AR_1$. Since $AR_1 \cap AR_2 = \emptyset$, $a \notin AR_2$; it follows by the fact that AR_2 is a locally-stable extension that there is $b \in AR_2$ such that $b \to a$. Therefore, v' has a child v'' owned by agent 1 such that the edge (v', v'') in $T_{\mathcal{A}}$ is labeled by b.

We complete the proof by noting that if v is the root, it has at least one child which is a decision node by the fact that $AR_1 \neq \emptyset$. □

We next explore the structure of $T_{\mathcal{A}}$ when $\mathcal{A}$ has AR_1 and AR_2 which satisfy some weaker flavor of defendability.

Proposition 17. *Let $i \in \{1, 2\}$, and $\mathcal{A}$ be a normal GBA framework such that AR_i is a locally-preferred extension and AR_{3-i} is locally-admissible. Then $\mathcal{A}$ is infinite.*

Before we tackle the proof, several lemmas are in order.

Lemma 18. *Let $\mathcal{A}$ be a normal GBA framework such that $AR_1 \cup AR_2$ is conflict-free. Then $AR_1 \cup AR_2$ is locally-admissible.*

Proof. The assumption that $AR_1 \cup AR_2$ is conflict-free directly implies that there is no $a \in AR_1 \cup AR_2$ and $b \in AR_1 \cup AR_2$ such that $a \to b$, and thus each $a \in AR_1 \cup AR_2$ is locally-acceptable. It follows that $AR_1 \cup AR_2$ is locally-admissible. □

Lemma 19. *Let $\mathcal{A}$ be a normal GBA framework such that AR_1 and AR_2 are both locally-admissible. In addition, assume that for all $a \in AR_1$ there is no $b \in AR_2$ such that $b \to a$. Then $AR_1 \cup AR_2$ is locally-admissible.*

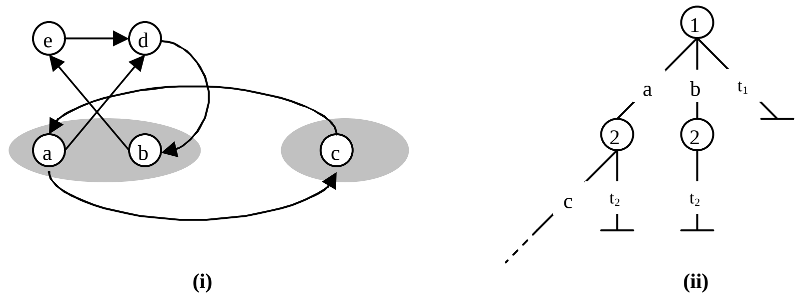


Figure 3: **(i)** $AR_1 = \{a, b\}$ and $AR_2 = \{c\}$ are nonempty, disjoint, and are both preferred extensions. **(ii)** $T_{\mathcal{A}}$ is not subgame-infinite.

Proof. We first show $AR_1 \cup AR_2$ is conflict free. Indeed, AR_1 and AR_2 are each conflict-free since they are locally-admissible. By the assumption, for all $a \in AR_1$ there is no $b \in AR_2$ such that $b \to a$. Furthermore, if there was $b \in AR_2$ and $a \in AR_1$ such that $a \to b$, we would get a contradiction to the local-admissibility of AR_2.

We complete the proof by applying Lemma 18. □

Lemma 20. *Let $\mathcal{A}$ be a normal GBA framework such that AR_1 and AR_2 are both locally-admissible. Then any tree rooted in a decision node of depth at least 2 of $T_{\mathcal{A}}$ must be infinite.*

Proof. Omitted due to lack of space. □

Proof of Proposition 17. Similarly to previous proofs, it is sufficient to show that at every level in the tree there is a decision node that has at least one child which is a decision node. For level 0 (the root), this follows from the non-emptiness of AR_1.

A setting where at level 1 every node has no children that are decision nodes directly implies that for all $a \in AR_1$ there is no $b \in AR_2$ such that $b \to a$. By Lemma 19, $AR_1 \cup AR_2$ is locally-admissible. Additionally, $AR_i \subsetneq AR_1 \cup AR_2$, since AR_{3-i} is nonempty and the two sets are disjoint. This is a contradiction to the assumption that AR_i is a locally-preferred extension.

The final case, where a decision node has depth of at least 2, is handled by Lemma 20. □

Remark 21. Requiring that $\mathcal{A}$ is normal and AR_1 and AR_2 be both admissible (and in particular, locally-admissible) does not entail that $\mathcal{A}$ is infinite. For example, if $AR = \{a, b\}$, $AR_1 = \{a\}$, $AR_2 = \{b\}$, and $\to = \emptyset$, then the conditions hold, but the height of $T_{\mathcal{A}}$ is 2.

Likewise, requiring that $\mathcal{A}$ is normal and that AR_1 and AR_2 be both preferred extensions does not entail that $\mathcal{A}$ is subgame-infinite (although this is less straightforward). A counterexample is given in Figure 3.

3.3 The Interaction Graph and the Structure of the Game Tree

The attack-defeat relations between the two rival agents in a GBA framework can be easily modelled as a two-sided graph.

Definition 22. Let $\mathcal{A} = \langle AR, \to, AR_1, AR_2, U \rangle$ be a GBA framework. The *interaction graph* of $\mathcal{A}$ is the directed bipartite graph $G_{\mathcal{A}} = \langle V, E \rangle$ where $V = AR_1 \uplus AR_2$, and

$$E = \{(v_1, v_2) : \ (v_1 \in AR_1 \wedge v_2 \in AR_2 \wedge v_2 \to v_1) \\ \vee (v_1 \in AR_2 \wedge v_2 \in AR_1 \wedge v_2 \to v_1)\}.$$

The definition of an interaction graph immediately entails a simple algorithm that decides whether a given argumentation framework is infinite.

Proposition 23. *A GBA framework $\mathcal{A}$ is infinite iff $G_{\mathcal{A}}$ is cyclic.*

Proof. Assume first that $\mathcal{A}$ is infinite. Since AR_1 is finite, there exists some path in $T_{\mathcal{A}}$ such that two edges e_1 and e_2 are labeled by an argument $a_1 \in AR_1$. Let $a_1, a_2, \ldots, a_k, a_1$ be the labels of the edges on the path. It holds that $a_i \in AR_1$ if i is odd, and $a_i \in AR_2$ if i is even. Moreover, $a_1 \to a_k \to \ldots \to a_2 \to a_1$, and therefore $a_1, a_2, \ldots, a_k, a_1$ is a cycle in $G_{\mathcal{A}}$.

In the other direction, suppose $G_{\mathcal{A}}$ is cyclic, and let $a_1, a_2, \ldots, a_k, a_1$ be a cycle in $G_{\mathcal{A}}$, where $a_i \in AR_1$ if i is odd, and is in AR_2 otherwise. Clearly, $a_1, a_2, \ldots, a_k, a_1, a_2, \ldots, a_k, a_1, \ldots$ are the labels of the edges in an infinite path in $T_{\mathcal{A}}$. □

Given a GBA framework, we can construct the interaction graph in polynomial time, and run breadth-first searches in order to determine whether cycles exist.[4] Therefore, it can be decided whether a given argumentation framework is infinite in polynomial time.

Remark 24. Proposition 23 also gives us a useful tool for deciding whether the conditions of Propositions 16 or 17 do not hold: this is the case if the argumentation framework is identified as finite.

3.4 Algorithmic Issues

It should now be quite apparent that we would like to model our arguments as finite games in extensive form. However, we have seen that when the argument sets AR_1 and AR_2 satisfy some flavors of defendability, the framework is guaranteed to be infinite.

We suggest two possible restrictions which make certain that $T_{\mathcal{A}}$ is always finite. One option is to require that each argument be usable only once. This is obviously sufficient, together with our veteran assumption that AR_1 and AR_2 are finite.

Another, more interesting, possibility is to restrict the length of dialogues. It is agreed beforehand that, in the event that neither agent terminates the argument until some stage, the debate automatically ends. This yields the concept of *k-bounded GBA frameworks*; these are the same as GBA frameworks, except for the fact that U also assigns a value to any legal dialogue of length k. Such dialogues terminate the argument even if they do not end in t_1 or t_2. Observe that the height of the game tree associated with a k-bounded GBA framework is at most k.

Given a finite GBA framework, the associated game in extensive form is a finite game of perfect information; such a game can be solved by backward induction [6], in time linear in the size of the game tree. Moreover, backward induction yields a sub-game perfect Nash equilibrium in pure strategies.

Definition 25. Let $\mathcal{A}$ be a finite GBA framework. The *outcome* of $\mathcal{A}$ is the payoff to the agents derived when solving $T_{\mathcal{A}}$ with backward induction.

Instead of engaging in a tiresome, and computationally costly, debate, the agents may simply and efficiently solve the corresponding extensive form game using backward induction. Since this method yields an exceptionally stable solution, both agents may readily agree on this method as an alternative to actually proceeding with the classic argumentation process.

Another method of simplifying a given (possibly infinite) GBA framework is the elimination of arguments which rational agents would never play in an argument. This approach might aid in reducing an infinite framework to a finite one.

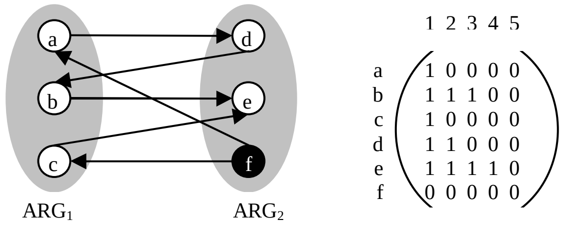
Definition 26. Let $\mathcal{A}$ be a normal GBA framework. For $i \in \{1, 2\}$, an argument $d \in AR_i$ is a *self-destructive* argument for agent i iff for every dialogue $D = (a_1, a_2, \ldots, a_l, d, t_{3-i})$ (some dialogue whose postfix is (d, t_{3-i})), the dialogue $D' = (a_1, a_2, \ldots, a_l, t_i)$ satisfies: $U^i(D') \geq U^i(D)$.

Proposition 27. *Let $i \in \{1, 2\}$ and d be a self-destructive argument for agent i. The outcome of the argument does not change if d is removed from AR_i.*

Proof. Omitted due to lack of space. □

Although the state of affairs looks promising at this point, being able to solve an argument in time linear in $T_{\mathcal{A}}$ may not imply encouraging results when considering the complexity with respect

[4] We might need a search for each node in the graph; each search's complexity is linear in the size of AR_1 and AR_2.



(i) **(ii)**

Figure 4: Illustration for the algorithm in the proof of Proposition 30. **(i)** $AR_1 = \{a, b, c\}$ and $AR_2 = \{d, e, f\}$, f is a winning argument for agent 2. **(ii)** The matrix which is constructed by the algorithm. If $k \geq 6$, agent 1 does not have a winning strategy.

to the size of the representation of $\mathcal{A}$. Granted, if the function U is represented by individually specifying the payoff for every possible dialog, the size of the representation of $\mathcal{A}$ would be on the order of the size of the tree. However, there are some cases where U is a simple function that can be represented with constant space. In such a setting, $T_{\mathcal{A}}$ might be exponential in the size of the representation of $\mathcal{A}$. We look at one such example, and discuss its efficient solution.

Definition 28. A normal k-bounded GBA framework $\mathcal{A}$ is a *pure* framework iff for any dialogue D such that $|D| \leq k$ and D ends with t_1 or t_2, $U(D) = (-0.5, 0.5)$ if agent 1 terminates the dialogue, and $U(D) = (0.5, -0.5)$ if agent 2 terminates the dialogue; and for any other dialogue D with $|D| = k$, $U(D) = (0.5, -0.5)$.

Remark 29. The representation of U in a pure GBA framework requires $O(1)$ space.

In the pure argumentation context, we say that an argument in AR_i is *winning* if it has no attackers in AR_{3-i}. The goal of agent 2 is to present a winning argument $w \in AR_2$ within less than k steps — thus forcing agent 1 to terminate the argument. Naturally, agent 1 can win in less than k steps by producing a winning argument of its own.

Proposition 30. *In a k-bounded pure GBA framework, it can be determined whether agent 1 has a winning strategy in time polynomial in $max(|AR_1|, |AR_2|)$ and k.*

Proof. We describe a dynamic programming algorithm which decides whether agent 1 has a winning strategy. Clearly it holds that agent 1 has a winning strategy iff agent 2 can be prevented from using any of its winning arguments within k steps.

We construct a binary matrix M of size $(|AR_1| + |AR_2|) \times (k - 1)$. Each row corresponds to an argument in $AR_1 \cup AR_2$; let $arg(i)$ be the argument associated with the i'th row. The (i, j) entry in the matrix M is 1 if, assuming a dialogue starts with $arg(i)$, agent 1 has a strategy that prevents agent 2 from playing a winning argument in j steps. The matrix is filled by columns. $M[i, 1]$ is initialized to 0 for each i where $arg(i)$ is a winning argument of agent 2; the rest of column 1 is initialized to 1. We assume agent 1 has no winning arguments — otherwise it can easily win the argument by playing its winning argument first. The other entries are computed as follows: $M[i, j] = 0$ if $M[i, j - 1] = 0$;

$$M[i, j] = min(M[i_1, j - 1], \ldots, M[i_l, j - 1])$$

if $M[i, j - 1] = 1$ and $arg(i) \in AR_1$ and $arg(i_1), \ldots, arg(i_l)$ are all the attackers of $arg(i)$ in AR_2;

$$M[i, j] = max(M[i_1, j - 1], \ldots, M[i_l, j - 1])$$

if $M[i, j - 1] = 1$ and $arg(i) \in AR_2$ and $arg(i_1), \ldots, arg(i_l)$ are all the attackers of $arg(i)$ in AR_1 (see example in Figure 4).

Less formally, if $arg(i) \in AR_2$, agent 1 has a strategy that prevents agent 2 from playing a winning argument in j steps if it can play an argument from AR_1, and then hold on for $j - 1$ steps against all possible responses from AR_2. Agent 2, on the other hand, wants to counter an argument in AR_1 with an argument which it knows will lead it to victory in $j - 1$ steps or less; it needs at least one such argument among the attackers of $arg(i)$.

The algorithm answers that agent 1 has a winning strategy iff there is i such that $arg(i) \in AR_1$ and $M[i, k-1] = 1$. It is straightforward that this is correct, since agent 1 can start the argument with $arg(i)$. As for the running time, the number of operations needed to compute an entry in the matrix M is $O(max(|AR_1|, |AR_2|)$, hence the running time is $O((|AR_1| + |AR_2|)^2 \cdot k)$.

□

4 Directions for Future Research

In this section we suggest two major directions for future research.

One possible extension of our work is to consider argumentation games with incomplete information. Such a setting is plausible, for example, when a mediator is involved in the argumentation process. A likely situation is one where agent 1 plays a valid, attacking argument, but does not want agent 2 to know which argument was played for security reasons; the mediator checks that the argument is valid, and informs agent 2 that its last argument was defeated. Agent 2 would then be in an information set which possibly contains several nodes. Such a game would be a two-player zero-sum (finite) extensive-form game of imperfect information but with perfect recall.[5] The equilibria of this game are the solutions of an LP whose size is linear in the size of the game tree [17]. Therefore, it seems interesting to study the properties of these argumentation systems, as well as define relevant, specialized semantics.

The second direction is a study of argumentation games where the agents are not familiar with one another's available arguments, i.e., agent 1 does not know which arguments are contained in AR_2, and agent 2 is ignorant about the members of AR_1. Another related setting is one where both agents are informed with respect to the argument sets, but have only partial knowledge of the attack relation. For example, agent 1 might not know which arguments in AR_2 attack an argument in AR_1.

5 Acknowledgments

The authors are grateful to an anonymous reviewer for useful comments. This work was partially supported by grant #039-7582 from the Israel Science Foundation.

References

[1] C. Cayrol, S. Doutre, and J. Mengin. On decision problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation*, 13(3):377–403, 2003.

[2] P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, 77(7):321–357, 1995.

[3] P. E. Dunne and T. J. M. Bench-Capon. Two party immediate response disputes: properties and efficiency. *Artificial Intelligence*, 149:221–250, 2003.

[4] P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, University of London, Queen Mary and Westfield College, Department of Electronic Engineering, 2000.

[5] S. Fatima, M. Wooldridge, and N. Jennings. Multi-issue negotiation under time constraints. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 143–150, 2002.

[6] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[7] W. Hodges. Logic and games. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2004. Available at: http://plato.stanford.edu/archives/win2004/entries/logic-games/.

[5] See [6] for the definition of perfect recall.

[8] H. Jakobovits and D. Vermeir. Dialectic semantics for argumentation frameworks. In *Proceedings of the Seventh International Conference on Artificial Intelligence and Law*, pages 53–62, 1999.

[9] R. Kowalczyk and V. Bui. On constraint-based reasoning in e-negotiation agents. In *Agent-Mediated Electronic Commerce III*, Lecture Notes in Computer Science, pages 31–46. Springer-Verlag, 2001.

[10] S. Kraus. *Strategic Negotiation in Multi-Agent Environments*. MIT Press, 2001.

[11] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[12] S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[13] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375, 2003.

[14] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, 1994.

[15] T. Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18(4):656–676, 2002.

[16] C. Sierra, N. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In *Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 1998.

[17] B. von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14:220–246, 1996.

[18] G. A. W. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Logics in Artificial Intelligence — Proceedings of the 7th European Workshop*, volume 1919 of *Lecture Notes in Computer Science*, pages 239–253. Springer-Verlag, 2000.

[19] M. Wooldridge. *An Introduction to Multiagent Systems*. Wiley, 2002.