A Formal Analysis of Dialogues on Infinite Argumentation Frameworks
Francesco Belardinelli, Davide Grossi, Nicolas Maudet

To cite this version:

HAL Id: hal-01167908
https://hal.archives-ouvertes.fr/hal-01167908
Submitted on 24 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Formal Analysis of Dialogues on Infinite Argumentation Frameworks

Francesco Belardinelli  
Laboratoire IBISC  
Université d’Evry, France  
belardinelli@ibisc.fr

Davide Grossi  
Department of Computer Science  
University of Liverpool, UK  
d.grossi@liverpool.ac.uk

Nicolas Maudet  
Sorbonne Universités  
UPMC Univ Paris 06  
CNRS, UMR 7606, LIP6  
F-75005, Paris, France  
nicolas.maudet@lip6.fr

Abstract

The paper analyses multi-agent strategic dialogues on possibly infinite argumentation frameworks. We develop a formal model for representing such dialogues, and introduce $FO_A$-ATL, a first-order extension of alternating-time logic, for expressing the interplay of strategic and argumentation-theoretic properties. This setting is investigated with respect to the model checking problem, by means of a suitable notion of bisimulation. This notion of bisimulation is also used to shed light on how static properties of argumentation frameworks influence their dynamic behaviour.

1 Introduction

The dialectical and dynamic dimensions of argumentation have been object of scrutiny since the inception of Dung’s abstract argumentation theory (cf. [Dung, 1994; 1995]). However, while the definition and analysis of ‘static’ justifiability criteria (i.e., argumentation semantics [Baroni et al., 2011]) has come to form the bulk of abstract argumentation theory, comparatively little work within Dung’s framework has been dedicated to a systematic study of forms of dynamic and multi-agent interaction. Some research has focused on operationalizations of argumentation semantics via two-player games (see [Modgil and Caminada, 2009] for an overview), while some other has attempted an analysis of strategic behavior in abstract forms of argumentation games (in particular [Procaccia and Rosenschein, 2005; Riveret et al., 2010; Thimm and Garcia, 2010]). This paper pursues further the understanding of multi-agent argumentation over abstract argumentation frameworks (AF) capitalizing on techniques from logic and multi-agent verification.

Contribution of the paper. The paper focuses on the formal analysis of multi-agent strategic interactions on possibly infinite argumentation frameworks. Agents are assumed to exchange arguments from possibly infinite AF. They hold private argumentation frameworks representing their ‘views’ on how arguments attack one another. They interact by taking turns and attacking relevant arguments expanding the framework underlying the interaction. This set up, which we call dynamic argumentation systems (DAS), is general enough to model a wide range of dialogue protocols and games on abstract AF. We analyse this setting formally by introducing $FO_A$-ATL, a novel first-order extension of the (turn-based) game logic ATL [Alur et al., 2002]. This logic allows us to specify dynamic properties of strategic interactions in argumentation. The objectives of the paper consist in: (i) the development of techniques to tackle the model-checking problem of $FO_A$-ATL; (ii) the development of techniques to analyse how static properties of argumentation frameworks influence their dynamic behavior. In other words, we are interested in isolating a class of AF for which their structure allows us to predict their dynamic behaviours. We provide preliminary positive results to both questions.

Technically, the paper models DAS as a special type of infinite-state data-aware systems [Deutsch et al., 2009; Hariri et al., 2013]. This allows us to capitalize on recent results on the formal verification of artifact-centric systems [Belardinelli et al., 2014], thus obtaining truth-preserving bisimulations for $FO_A$-ATL.

Related work. The paper contributes to several current strands of research in abstract argumentation.

Dynamics of argumentation. How argumentation framework may change by performing operations on their structure has been object of several recent works (e.g., [Baumann, 2012; Bisquert et al., 2013; Booth et al., 2013; Doutre et al., 2014]). However, all mentioned papers assume finite AF, as they rely on the possibility of encoding them into propositional formulas [Besnard and Doutre, 2004]. Neither do they incorporate agency, as they analyse argumentation dynamics from a belief revision paradigm. Our contribution relaxes the finiteness assumption and models agents explicitly as ‘protocols’ dynamically modifying the structure of AF.

Infinite argumentation frameworks. The bulk of known results in abstract argumentation typically pertain to finite AF. However, infinite AF are gaining attention and have been object of several recent contributions [Baroni et al., 2012; 2013; Baumann and Spanring, 2015], which essentially focus on how known results for the finite case generalize to the infinite. Allowing an infinity of arguments is critical in applications where upper bounds on the number of available arguments cannot be established a priori. Our paper contributes to the understanding of infinite AF.

Logics for abstract argumentation. Recently, several formulations of argumentation theory have been put forward.
(e.g., [Caminada and Gabbay, 2009; Grossi, 2010] for early contributions). These works typically focus on finding logical languages (from modal to many-valued logics) that are sufficiently expressive to represent argumentation semantics. Our focus here is rather to specify the strategic abilities of agents engaging in a dialogue/dispute, such as: ‘the proponent is able to respond to all attacks by maintaining a conflict-free set of arguments’ or ‘the opponent has a strategy to force proponent to run out of arguments’. In this respect the paper is a first contribution to the specification of multi-agent argumentation frameworks by means of temporal logics, as well as to their formal verification (cf. [Lomuscio et al., 2009]).

**Outline of the paper.** Section 2 introduces the dynamics of (multi-agent) argumentation frameworks and the specification language FO\(A\)-ATL; then we state the corresponding model checking problem. Section 3 contains the main technical result, namely, bisimilar dynamic argumentation systems (DAS) satisfy the same formulas in FO\(A\)-ATL. We also investigate the impact of static features of DAS on their dynamics. We conclude in Section 4 and point to future work. For reasons of space all proofs are omitted.

2 The Dynamics of AF

In this section we introduce dynamic argumentation systems (DAS). Then, we present FO\(A\)-ATL, a first-order version of the alternating-time temporal logic ATL [Alur et al., 2002], and state the corresponding model checking problem. We first present the basic terminology to be used in the paper.

2.1 Abstract Argumentation

The theoretical setting is built on abstract argumentation, as introduced in [Dung, 1994; 1995]. In what follows we assume a finite set \(Ag = \{a_0, \ldots , a_n\}\) of names for agents.

**Definition 1 (Argumentation Framework)** Given a set \(Ag\) of agent names, a (multi-agent) argumentation framework is a tuple \(A = (Ag, \prec_a)\) s.t. (i) \(A\) is a (possibly infinite) set of arguments, and (ii) for every \(a \in Ag\), \(\prec_a \subseteq A^2\) is an attack relation between arguments.

Notice that Def. 1 allows argumentation frameworks, or AF, that include infinitely many arguments. This choice, while providing greater modelling flexibility, also reflects the fact that upper bounds on the number of arguments available to each agent cannot be easily established in general.

We define \(F(A, Ag)\) as the set of all AF built on sets \(A\) of arguments and \(Ag\) of agent names. Hereafter, we simply write \(F(Ag)\) whenever \(A\) is clear. Also, given set \(Ag = \{p, o\}\) containing the proponent \(p\) and opponent \(o\), \(F(p)\) (resp. \(F(o)\)) are shorthands for \(F(\{p\})\) (resp. \(F(\{o\})\)), i.e., the set of AF for proponent \(p\) (resp. opponent \(o\)). Finally, we define the (unindexed) attack relation \(\leftarrow = \bigcup_{a \in Ag} \prec_a\).

To express relevant properties of AF, we introduce FO\(A\), a first-order language geared towards expressing properties of attack relations. Let \(Ag\), for \(a \in Ag\), be binary predicate symbols, \(P_0, P_1, \ldots\) unary predicate symbols, and \(\text{Var}\) a countable set of individual variables. We define first-order formulas \(\varphi\) and the set \(fr(\varphi)\) of free variables by mutual recursion.

**Definition 2 (FO\(A\))** The formulas \(\varphi\) and the set of free variables \(fr(\varphi)\) are defined as follows:

- if \(P\) is a predicate symbol and \(x \in \text{Var}\) a variable, then \(P(x)\) is a formula with \(fr(P(x)) = \{x\}\);
- if \(\varphi\) is a formula, then also \(\neg \varphi\) is, and \(fr(\neg \varphi) = fr(\varphi)\);
- if \(\varphi\) and \(\varphi'\) are formulas with \(fr(\varphi) = fr(\varphi')\), then also \(\varphi \rightarrow \varphi'\) is, and \(fr(\varphi \rightarrow \varphi') = fr(\varphi) \times fr(\varphi')\);
- if \(\varphi\) is a formula with \(fr(\varphi) = \{y\}\), then also \(\forall y(A_y(x, y) \rightarrow \varphi)\) and \(\forall y \varphi\) are, with \(fr(\forall y(A_y(x, y) \rightarrow \varphi)) = \{x\}\) and \(fr(\forall y \varphi) = \emptyset\).

In the following we use the standard abbreviations \(\land, \lor,\) and \(\exists\). Specifically, \(\exists y(A_y(x, y) \land \varphi)\) (resp. \(\exists y \varphi\)) is a shorthand for \(\neg \forall y(A_y(x, y) \rightarrow \varphi)\) (resp. \(\neg \forall y \varphi\)). Notice that every FO\(A\) formula \(\varphi\) has at most one free variable.

Finally, we extend argumentation frameworks with interpretations \(\pi\) that assign a subset of \(A\) to each predicate symbol \(P\), i.e., \(\pi(P) \subseteq A\). An interpreted argumentation framework is defined as a couple \((A, \pi)\). We can now introduce the semantics of our first-order-language.

**Definition 3 (Semantics of FO\(A\))** We define whether an argument \(u \in A\) satisfies an FO\(A\)-formula \(\varphi\) in an interpreted AF \((A, \pi)\), or \((A, \pi, u)\) \(\models \varphi\), as follows (for clauses (for propositional connectives are straightforward, thus omitted):

\[
(\forall, \pi, u) \models P(x) \quad \text{iff } u \in \pi(P)
\]
\[
(\forall, \pi, u) \models \forall y(A_y(x, y) \rightarrow \varphi) \quad \text{iff for every } v \in A,
\]
\[
u \rightarrow_\pi v \text{ implies } (\forall, \pi, v) \models \psi
\]
\[
(\forall, \pi, u) \models \forall y \varphi \quad \text{iff for every } v \in A, (\forall, \pi, v) \models \psi
\]

An FO\(A\)-formula \(\varphi\) is true in an interpreted AF \((A, \pi)\), or \((A, \pi)\) \(\models \varphi\), if \((A, \pi, u)\) \(\models \varphi\) for every argument \(u \in A\); \(\varphi\) is valid in \(A\), or \(A\) \(\models \varphi\), if \((A, \pi)\) \(\models \varphi\) for every interpretation \(\pi\).

From a technical point of view, the language FO\(A\) is the dyadic fragment of first-order logic with one free variable.\(^1\) Table 1 shows how this simple logic suffices to formalize several of the key notions from [Dung, 1995] (see also [Grossi, 2010]).

2.2 Dynamic Argumentation Frameworks

In this paper we are interested in analysing dialogues modeled as multi-agent processes over argumentation frameworks. As dialogues progress agents build a ‘shared’ (or ‘public’) argumentation framework which is used to evaluate the available arguments.

**Agents and global states**

To introduce the dynamics of argumentation frameworks we start with the notion of agent.

**Definition 4 (Agent)** Given sets \(Ag\) of agent names and \(A\) of arguments, an agent is a tuple \(a = (Ag, Act, Pr)\) where

- \(A \in F(a)\) is the agent’s argumentation framework;
- the set Act of actions contains action \(\text{attack}(x, x')\) to attack argument \(x'\) with argument \(x\), and action \(\text{skip}\);

\(^1\) As such, it is equivalent to the multi-modal logic K with the universal modality [Blackburn et al., 2001].
\( \pi(\mathcal{A}) \) is conflict-free in \( \mathcal{A} \) iff \( (\mathcal{A}, \pi) \models \forall x(P(x) \rightarrow \neg(\exists y(A(y, x) \land P(y)))) \)

\( \pi(\mathcal{A}) \) is acceptable in \( \mathcal{A} \) iff \( (\mathcal{A}, \pi) \models \forall x(P(x) \rightarrow \forall y(A(y, x) \rightarrow \exists z(A(z, y) \land P(z)))) \)

\( \pi(\mathcal{A}) \) is admissible in \( \mathcal{A} \) iff \( (\mathcal{A}, \pi) \) is conflict-free and acceptable

\( \pi(\mathcal{A}) \) is complete in \( \mathcal{A} \) iff \( (\mathcal{A}, \pi) \) is conflict-free and \( (\mathcal{A}, \pi) \models \forall x(P(x) \leftrightarrow \forall y(A(y, x) \rightarrow \exists z(A(z, y) \land P(z)))) \)

\( \pi(\mathcal{A}) \) is a stable in \( \mathcal{A} \) iff \( (\mathcal{A}, \pi) \models \forall x(P(x) \leftrightarrow \neg(\exists y(A(y, x) \land P(y)))) \)

Table 1: Some key notions of abstract argumentation theory and their formalization in FO\(_A\) (\( A \) is interpreted over \( \rightarrow \)).

- **Pr**: \( \bigcup_{\mathcal{A} \subseteq \mathcal{A}} \mathcal{F}(\mathcal{A}, A_g) \rightarrow 2^{Act(A)} \) is the local protocol function, where \( Act(A) \) is the set of ground actions \( \alpha(i, \bar{i}) \), and for \( i \) \( \in \mathcal{F}(\mathcal{A}, A_g) \), \( attack(u, u') \in Pr(\mathcal{A}) \) only if \( u' \in \mathcal{A} \) and \( u' \leftarrow u \) holds in \( \mathcal{A} \); while the \textit{skip} action is always enabled.

By Def. 4 agent \( a \)'s local state is modelled as an argumentation framework \( \mathcal{A} \), and she is assumed to perform the actions in \( Act \) according to protocol \( Pr \). In particular, by definition of \( Pr \), attacks must be relevant, i.e., the attacked argument \( u' \) has to appear in the current argumentation framework \( \mathcal{A} \), and they must reflect actual attacks available in the agent’s AF \( \mathcal{A} \in \mathcal{F}(a) \). These assumptions on protocols model the basic requirements of relevance and truthfulness in dialogues.

To model agent interaction we need to define their composition, beginning with the notion of global state.

**Definition 5 (Global State)** Given a set \( Ag \) of agents \( a \in \{A_i, Act_i, Pr_i\} \) defined on the same (possibly infinite) set \( A \) of arguments, a global state is a couple \((s, a)\) where (i) \( s \) \( \in \mathcal{F}(\mathcal{A}, A_g) \) is an argumentation framework for some \( \mathcal{A} \subseteq A \); and (ii) \( a \in Ag \).

In Def. 5 all agents are defined on the same set of arguments. However, some literature on argumentation theory suppose that each agent is endowed with a distinct set of arguments (e.g., [Rahwan and Larson, 2011]). We remark that our requirement is not a limitation because, firstly we can always consider the union of the sets of arguments for each agent, and secondly the characterising feature of AF is really the attack relation. Moreover, this assumption simplifies the presentation hereafter. Also, in a state \((s, a)\), \( a \) is the active agent, i.e., the agent to act next; it will be omitted whenever clear from the context. We write \( \mathcal{G} \) to denote the set of all global states. Finally, any set \( \mathcal{A} \subseteq A \) can be seen as a global state where, for every \( a \in Ag \), \( \leftarrow a \) is empty.

**Argument Dynamics**

We now introduce the dynamics of argumentation frameworks. In the rest of the paper we suppose that the only two agents are proponent \( p \) and opponent \( o \). This assumption can be lifted but it crucially simplifies the presentation of the key (conceptual and technical) contributions of the paper.

**Definition 6 (DAS)** Given set \( Ag = \{p, o\} \) of agents defined on the same (possibly infinite) set \( A \) of arguments, a dynamic argumentation system is a tuple \( \mathcal{P} = \{Ag, I, \tau, \pi\} \) where

- \( I \subseteq A \times \{o\} \) is the set of initial global states \((s_0, o)\);
- \( \tau : \mathcal{G} \times (Act_p(A) \cup Act_o(A)) \rightarrow \mathcal{G} \) is the transition function, where \( \tau((s, a), attack_a(u')) \) is defined iff \( a = a' \text{ and } attack_a(u') \in Pr_a(s) \). Moreover, \( (s', a') = \tau((s, a), attack(u, u')) \) iff \( a' \neq a \) and \( s' = (\mathcal{A}', \leftarrow') \)

for \( \mathcal{A}' = A \cup \{u\} \) and \( \leftarrow' = \leftarrow \cup \{(u', u)\} \); while \( (s', a') = \tau((s, a), \text{skip}) \) iff \( a' = a \) and \( s' = s \).

\( \pi \) is an interpretation of predicate symbols \( P \) as above.

A DAS evolves from an initial state \((s_0, o) \in I\) as specified by the transition function \( \tau \), which returns a successor state for the current state and selected action. DAS are turn-based, with each agent taking one action at a time. Hereafter we assume that the opponent always moves first, that is, \( I \) is intuitively the set of arguments supported by the proponent. One key feature of DAS is that, since the set \( A \) of arguments is infinite in general, they are infinite-state systems normally. Indeed, as the dialogue unfolds, agents can introduce an infinite number of arguments from \( A \), thus going through infinitely many states during the system’s execution.

We now fix some notation. The \textit{transition relation} \( \rightarrow_a \) is defined so that \( s \rightarrow_a s' \) iff for some action \( \alpha_a(u) \in Act_a(A) \), for \( a \in \{o, p\} \), we have that \( s \xrightarrow{\alpha_a(u)} s' \), i.e., \( \tau((s, a), \alpha_a(u)) = (s', a') \) with \( a' \neq a \). A run \( \lambda \) from a state \((s, a)\), or \((s, a)\)-run, is an infinite sequence \( s^0 \rightarrow_a s^1 \rightarrow_a \cdots \) s.t. \( s^0 = s \). Notice that, since agents \( o \) and \( p \) take turns in performing actions, the sequence of agents in an \((s, a)\)-run is uniquely determined by \( a \), so we omit agents whenever these are clear by the context. Hence, we will often denote a state \((s, a)\) simply as \( s \), whenever agent \( a \) is understood.

For \( n, m \in \mathbb{N} \), with \( n \leq m \), we define \( \lambda(n) = s^n \) and \( \lambda[n, m] = s^n, s^{n+1}, \ldots, s^m \). A state \( s' \) is \textit{reachable} from \( s \) iff for some \( s\)-run \( \lambda \) and \( i \geq 0 \), \( s' = \lambda(i) \). Let \( \mathcal{S} \) be the set of states reachable from any initial state \( s_0 \in I \). Notice also that the transition relation is serial, as each agent has a \textit{skip} action enabled at each local state.

**DAS, dialogues and games**

It is worth comparing DAS briefly with known structures in the literature on dialogue games for abstract argumentation. In the terminology of [Thimm and Garcia, 2010] DAS can be viewed as defining a dialectical game protocol, where at each step an active agent attacks some of the available arguments. DAS can be seen as extensive game forms with infinite horizon (though they are not games, as they do not specify payoffs). They also generalize the structures underpinning the extensive-form argument games of [Procaccia and Rosenschtein, 2005] by allowing agents to hold an infinite argumentation framework and to attack any of the available arguments, not only the last one uttered by some other agent.

We finally show how standard dialogue games (e.g., for the grounded extension [Dung, 1994] or for credulous admissibility [Vreeswijk and Prakken, 2001]) can be modeled as DAS.

**Example 1** In the game for the grounded extension two
agents \( o \) and \( p \) hold the same private argumentation framework, i.e., \( \mathcal{A}_o = \mathcal{A}_p \). These games are two-player zero-sum perfect information games with possibly infinite horizon. We can view them as DAS. Consider for instance the AF in Figure 1. The DAS corresponding to the game for grounded played on this AF, starting at argument \( t_1 \) can be defined as follows. Fix \( I = \{ t_1 \} \). For both agents we can define the following protocol: if the current framework contains \( t_i \) then attack \( t_i \) with \( u_i \) or \( t_{i+1} \), otherwise \( \text{skip} \). This protocol encodes the ‘legal’ moves of \( o \) and \( p \) according to the game for the grounded extension. The possible runs of this DAS contains all possible sub-graphs of the above AF generated from \( t_i \), e.g.: \( \{ \{ t_1 \}, \emptyset \} \), \( \{ \{ t_1, u_1 \}, \{ u_1, t_1 \} \} \), \( \{ \{ t_1, t_2 \}, \{ t_2, t_1 \} \} \), \( \{ \{ t_1, t_2 \}, \{ t_2, t_1 \}, \{ t_1, t_1 \} \} \) \ldots . These runs correspond to possible dialogues in the game for the grounded extension.

Given a DAS \( P \) on a (possibly infinite) set \( A \) of arguments, we define the corresponding (joint) argumentation framework \( \mathcal{A}_P = \langle A, \{ \rightarrow_a \}_{a \in A} \rangle \) so that \( u \rightarrow_a \rightarrow u' \) holds in \( \mathcal{A}_P \) iff \( u \rightarrow_a \rightarrow u' \) holds in the AF \( \mathcal{A}_a \) for agent \( a \in A \). By the definition of protocols (Def. 4) it is apparent that every reachable state in \( \mathcal{A}_P \) is a subgraph of \( \mathcal{A}_a \), that is,

\[
\text{if } u \rightarrow_a \rightarrow v \text{ holds in some } s \in S, \text{ for some } a \in A, \text{ then } u \rightarrow_a \rightarrow v \text{ holds in } \mathcal{A}_a. (*)
\]

Thus, states in \( \mathcal{P} \) are truthful, yet partial, representations of \( \mathcal{A}_P \). However, the converse of (\( * \)) does not hold in general, i.e., \( \mathcal{P} \) needs not to include all subgraphs of \( \mathcal{A}_P \) as states. To appreciate this consider the following:

**Example 2** [Example 1 continued] Now endow \( o \) with a more restrictive protocol: if the current framework contains \( t_i \) then attack \( t_i \) with \( u_i \), otherwise \( \text{skip} \). This protocol makes \( o \) play more rationally, selecting arguments to which \( p \) cannot reply. The only possible run of this DAS is:

\[
\{ \{ t_1 \}, \emptyset \}, \{ \{ t_1, u_1 \}, \{ u_1, t_1 \} \}, \text{ skip}, \ldots .
\]

This asymmetry motivates the next definition. An agent is \textit{naive} if its protocol allows it to move any available attack (cp. this to the more restrictive protocol of \( o \) in Example 1).

**Definition 7 (Naive Agent)** An agent \( a \) is \textit{naive} \( \iff \) for every \( A' \in F(A', Ag) \), \( \text{attack}(u,u') \in \text{Pr}(A') \ \iff \ u' \in A' \text{ and } u' \rightarrow_a u \text{ holds in } A_a \).

Intuitively, naive agents perform every possible attack available to them, irrespectively of the strategic behaviour of other agents. A DAS \( \mathcal{P} \) is \textit{naive} \( \iff \) every agent in \( Ag \) is naive. Then, we remark that for naive DAS the converse of (\( * \)) actually holds.

### 2.3 The Specification Language \( \mathcal{F}_A \)-ATL

We now introduce a formal language to specify properties of interest of DAS. Arguments in DAS call for the use of first-order logic, whereas strategy operators are needed to account for agents’ behaviour. To the best of our knowledge, no logic combining these two features has ever been studied.

**Definition 8 (\( \mathcal{F}_A \)-ATL)** The \( \mathcal{F}_A \)-ATL formulas \( \varphi \), with free variables \( \text{fr}(\varphi) \), are defined as follows:

- if \( \psi \) is an \( \mathcal{F}_A \)-formula, then it is an \( \mathcal{F}_A \)-ATL formula;
- if \( \varphi \) is a formula and \( N \subseteq Ag \), then \( \neg \varphi \) and \( (\neg)\varphi \) are formulas with \( \text{fr}(\neg \varphi) = \text{fr}(\neg)\varphi \);
- if \( \varphi \) and \( \varphi' \) are formulas with \( \text{fr}(\varphi) = \text{fr}(\varphi') \) and \( N \subseteq Ag \), then \( \varphi \rightarrow \varphi' \) and \( (\neg)\varphi \varphi' \) are formulas with \( \text{fr}(\varphi \rightarrow \varphi') = \text{fr}(\neg)\varphi \varphi' \) = \( \text{fr}(\varphi) = \text{fr}(\varphi') \);
- if \( \varphi \) is a formula with \( \text{fr}(\varphi) = \{ y \} \), then \( \forall y (\text{Ag}(y, x) \rightarrow \varphi) \) and \( \forall y \varphi \) are formulas with \( \text{fr}(\forall y (\text{Ag}(y, x) \rightarrow \varphi)) = \{ x \} \) and \( \text{fr}(\forall y \varphi) = \emptyset \).

The language \( \mathcal{F}_A \)-ATL is a first-order extension of the alternating-time temporal logic \( \mathcal{ATL} \) [Altur et al., 2002]. The \( \mathcal{F}_A \)-ATL formulas (i) \( (\neg)\varphi \), (ii) \( (\neg)\varphi \varphi' \) and (iii) \( (\neg)\varphi \varphi' \) are read as ‘the agents in \( N \) have a strategy to...’ (i) ‘...enforce \( \varphi \) at the next state’, (ii) ‘...always enforce \( \varphi \)’, and (iii) ‘...enforce \( \varphi \) until \( \psi \).’

To interpret \( \mathcal{F}_A \)-ATL formulas on DAS we need to introduce the notion of a \textit{strategy} for a set \( N \) of agents. An \textit{N}-strategy is a mapping \( f_N : S^+ \rightarrow \bigcup_{a \in N} \text{Act}_a(A) \) s.t. \( f_N(k \cdot (s, a)) \in \text{Pr}_a(s) \) for every \( k \in S^* \). Intuitively, a strategy returns an enabled action in \( \text{Act}_a(A) \) for every non-empty, finite sequence of states in \( S^+ \). We remark that, according to standard terminology in concurrent game models [Bulling et al., 2010], the agents in DAS have \textit{perfect information} and \textit{perfect recall}, that is, their strategies are determined by all information available at each global state, for all states visited up to the current state. Further, the \textit{outcome} of strategy \( f_N \) at state \( (s, a) \), or \( (s, f_N) \), is the set of all \( (s, a) \)-runs \( \lambda \) s.t. for every \( b \in N \), \( \lambda(i+1), b' = \tau((\lambda(i), b), f_N(\lambda(i), i)) \) for all \( i \geq 0 \). As above, we assume that strategies respect agents’ turns and simply write \( (s, f_N) \), thus omitting agents whenever the latter are clear by the context.

**Definition 9 (Semantics of \( \mathcal{F}_A \)-ATL)** We define whether an argument \( u \) satisfies a formula \( \varphi \) at state \( s \) in a DAS \( \mathcal{P} \), or \( \langle \mathcal{P}, s, u \rangle \models \varphi \), as follows (clauses for propositional connectives are straightforward and thus omitted):

\[
\langle \mathcal{P}, s, u \rangle \models \psi \iff (s, \pi, u) = \psi, \text{ if } \psi \text{ is an } \mathcal{F}_A \text{-formula}
\]

\[
\langle \mathcal{P}, s, u \rangle \models (\neg)\varphi \iff \text{for some N-strategy } f_N, \text{ for all } \lambda \in \text{out}(s, f_N), \langle \mathcal{P}, \lambda(1), u \rangle \models \varphi
\]

\[
\langle \mathcal{P}, s, u \rangle \models (\neg)\varphi \varphi' \iff \text{for some N-strategy } f_N, \text{ for all } \lambda \in \text{out}(s, f_N), \text{ for some } k \geq 0, \langle \mathcal{P}, \lambda(k), \varphi \rangle, \langle \mathcal{P}, \lambda,j, u \rangle \models \varphi
\]

\[
\langle \mathcal{P}, s, u \rangle \models \forall y (\text{Ag}(y, x) \rightarrow \varphi) \iff \text{for every } v \in s, u \rightarrow_a v \text{ implies } \langle \mathcal{P}, s, v \rangle \models \varphi
\]

\[
\langle \mathcal{P}, s, u \rangle \models \forall y \varphi \iff \text{for every } v \in s, \langle \mathcal{P}, s, v \rangle \models \varphi
\]

A formula \( \varphi \) is true at \( s \), or \( \langle \mathcal{P}, s \rangle \models \varphi \), if \( \langle \mathcal{P}, s, u \rangle \models \varphi \) for every arguments \( u \in s \); \( \varphi \) is true in \( \mathcal{P} \), or \( \mathcal{P} \models \varphi \), if \( \langle \mathcal{P}, s_0 \rangle \models \varphi \) for all \( s_0 \in I \).
We illustrate now the expressiveness of FO₄-ATL through some example properties that involve strategic and argumentation-theoretic features.

**Example 3** The following formula states that opponent \( o \) can force proponent \( p \) to run out of moves in the next state:

\[
\langle o \rangle X \forall x \neg \exists y A_p(y, x)
\]

This formula is for instance true at argument \( t_1 \) in the DAS of Example 1.

**Example 4** The following formula states that proponent \( p \) has a strategy enforcing the set of arguments in \( P \), which includes the current argument, to be conflict-free (respectively, acceptable, admissible, complete, stable):

\[
P(x) \land \langle p \rangle G \chi(P)
\]

where \( \chi \in \{Cfr, Acc, Adm, Cmp, Stb\}. \) Let us focus for instance on \( P(x) \land \langle p \rangle G \text{Adm}(P) \). This is a desirable requirement for proponent’s strategies in dialogue games for the grounded extension (recall Example 1) or credulous admissibility [Vreeswijk and Prakken, 2000]. This statement is false of the DAS of Example 1 (at argument \( t_1 \)) where \( o \), by moving \( \text{attack}(u_1, t_1) \), forces \( p \) to run out of moves, no matter what interpretation \( \pi \) of \( P \) is selected. In fact \( p \) has no strategy to force any set of arguments containing \( t_1 \) (the initial argument) to be admissible. That is: \( P(x) \rightarrow \langle o \rangle G \neg \text{Adm}(P) \) or, equivalently, \( P(x) \rightarrow \neg \langle p \rangle G \text{Adm}(P) \) for any \( P \). □

### 3 Verifying Dynamic Argumentation Systems

In Section 2 we introduced DAS as a general model for (abstract) argument-based multi-agent interaction. We tailored an influential game logic in order to provide DAS with the specification language FO₄-ATL. We are then able to define a model checking problem whereby strategic and argumentation-theoretic properties, such as those expressed in (1) and (2) above, can be checked against a given DAS.

**Definition 10 (Model Checking Problem)** Given a DAS \( P \) and an FO₄-ATL-formula \( \varphi \), determine whether for every \( s_0 \in I, (P, s_0, u_0) \models \varphi \) for some argument \( u_0 \).

In case that \( \varphi \) is a sentence with no free variable, the model checking problem reduces to verify whether \( P \models \varphi \).

Model checking general data-aware systems is known to be undecidable [Deutsch et al., 2007]. In [Belardinelli et al., 2012; 2014] the same problem is proved decidable for bounded and uniform systems. Without going into details, we remark that DAS do not normally satisfy these requirements. Therefore new techniques need to be developed.

In this section we introduce a notion of bisimulation to explore under which circumstances two DAS satisfy the same formulas. In particular, we show that bisimilar DAS satisfy the same FO₄-ATL formulas. Then, we investigate how the static properties of AF impact on the dynamics of DAS. This kind of results is key to tackle the model checking problem, as they allow to verify a DAS \( P \) by model checking a bisimilar DAS \( P' \).

### 3.1 Bisimulations for Argument Frameworks

A notion of bisimulation can naturally be defined on AF (cf. [Grossi, 2010]).

**Definition 11 (Static Bisimulation)** Let \( (A, \pi) = (A, \{a_{Ag}, \pi\}) \) and \( (A', \pi') = (A', \{a_{Ag}, \pi'\}) \) be interpreted AF defined on a set \( Ag \) of agents. A static bisimulation is a relation \( S \subseteq A \times A' \) s.t. for \( u, v \in A, u' \in A' \), \( S(u, u') \) implies:

1. for every predicate symbol \( P \), \( u \in \pi(P) \) iff \( u' \in \pi'(P) \);
2. for every \( v \in A \), if \( u \leftarrow a \ v \) then for some \( v' \in A' \), \( u' \leftarrow a \ v' \) and \( S(v, v') \);
3. for every \( v' \in A' \), if \( u' \leftarrow a \ v' \) then for some \( v \in A \), \( u \leftarrow a \ v \) and \( S(v, v') \).

Two arguments \( u \in A, u' \in A' \) are bisimilar, or \( u \simeq u' \), iff \( S(u, u') \) for some static bisimulation \( S \). Finally, two AF \( A \) and \( A' \) are statically bisimilar (or simply bisimilar) iff (i) for every \( u \in A, u \simeq u' \) for some \( u' \in A' \); and (ii) for every \( u' \in A', u' \simeq u \) for some \( u \in A \). We denote this as \( A \simeq A' \).

We can now show that bisimilar states satisfy exactly the same FO₄-formulas.

**Lemma 1** Given bisimilar interpreted AF \( (A, \pi) \) and \( (A', \pi') \), and bisimilar arguments \( u \in A \) and \( u' \in A' \), then for every FO₄-formula \( \varphi \),

\[
(A, \pi, u) \models \varphi \iff (A', \pi', u') \models \varphi
\]

As a result, bisimilar AF cannot be distinguished by FO₄-formulas. In the following we explore the conditions under which this applies to DAS as well. In particular, we say that DAS \( P \) and \( P' \) are statically bisimilar iff \( A_P \simeq A_{P'} \) for some static bisimulation that maps initial states into initial states.

### 3.2 Bisimulations for DAS

We first introduce a notion of dynamic bisimulation, and then explore its properties in the context of DAS. In the rest of the section we let \( P = (Ag, I, \pi, \tau) \) and \( P' = (Ag, I', \tau', \pi') \) be two DAS defined on the same set \( Ag \) of agent names. Notice that, albeit agents may have the same name, they might differ as to their argumentation frameworks, actions, or protocols.

**Definition 12 (Dynamic Simulation)** Given DAS \( P \) and \( P' \), a dynamic simulation is a relation \( R \subseteq S \times S' \) s.t. for \( s \in S, s' \in S' \), \( R(s, s') \) implies:

1. \( s \approx s' \) for some static bisimulation \( S \);
2. for every \( t \in S \), if \( s \rightarrow_a t \) then for some \( t' \in S' \), \( s' \rightarrow_a t' \), \( t \approx t' \) for some bisimulation \( S' \subseteq S \), and \( R(t, t') \).

In Def. 12 we implicitly assume that the simulation relation relates states with turns for the same active agent in \( Ag \). Simulations can then be naturally extended to bisimulations.

**Definition 13 (Dynamic Bisimulation)** A relation \( D \subseteq S \times S' \) is a dynamic bisimulation iff both \( D \) and \( D^{-1} = \{(s', s) \mid D(s, s')\} \) are dynamic simulations.

Two states \( s \in S \) and \( s' \in S' \) are dynamically bisimilar, or \( s \approx s' \), iff \( D(s, s') \) for some bisimulation relation \( D \). It can be shown that \( \approx \) is the largest dynamic bisimulation, and an equivalence relation, on \( S \cup S' \). DAS \( P \) and \( P' \) are dynamically bisimilar, or \( P \simeq P' \), iff (i) for every \( s_0 \in I, s_0 \approx s'_0 \)

---

2Recall the properties in Table 1.

3It is a known validity of turn-based asynchronous ATL that \( \langle N \rangle G \phi \leftrightarrow \langle Ag \setminus N \rangle \neg \phi \) [Alur et al., 2002].
In some specific cases there is indeed a correspondence. The static properties of DAS determine their dynamic features. In Lemma 2 we showed that this relationship is not straightforward, not even in the case of naive DAS. However, in some specific cases there is indeed a correspondence. The first result reduces dynamic bisimulations to static bisimulations, together with some constraints on temporal transitions. As usual, we assume that turns are respected.

**Theorem 4** Let \( \mathcal{P} \) and \( \mathcal{P}' \) be DAS. Suppose that \( \mathcal{P}' \) is naive and for every \( u \in S, u' \in S' \), if \( s \simeq s' \), \( u \simeq u' \) w.r.t. \( s \) and \( s' \), and \( u \leftarrow_a v \) in \( \mathcal{P} \) for some \( v \in \mathcal{A} \), then \( u' \leftarrow_a v' \) in \( \mathcal{P}' \) for some \( v' \in \mathcal{A}' \) and either

1. \( v \in s \) and either (i) \( v' \in s' \) and \( v \simeq v' \) w.r.t. \( s \) and \( s' \), or (ii) \( v' \notin s' \) and for no \( w \in S, v \leftarrow_a w \) in \( s' \).
2. or \( v \notin s \) and either (i) \( v' \notin s' \), or (ii) \( v' \in s' \) and for no \( w \in s' \), \( v' \leftarrow_a w \) in \( s' \).

Then, \( D = \{(s, s') | s \simeq s'\} \) is a dynamic simulation between \( \mathcal{P} \) and \( \mathcal{P}' \).

Notice that the assumptions in Theorem 4 imply that DAS \( \mathcal{P} \) and \( \mathcal{P}' \) are statically bisimilar. However, they have also to satisfy extra conditions (1) and (2).

The main contribution of Theorem 4 is to show that the notion of dynamic bisimulation can be reduced to static bisimulation, together with some assumptions on the structural properties of AF \( \mathcal{A}_P \) and \( \mathcal{A}_P' \), in case that we consider naive DAS. Hence, in order to verify DAS we can simply model check statically bisimilar systems, and then transfer the result by using Theorem 4.

It is of interest to analyse DAS that actually satisfy the conditions above. For example, it is easy to check that whenever the underlying AF \( \mathcal{A}_P \) and \( \mathcal{A}_P' \) of naive DAS \( \mathcal{P} \) and \( \mathcal{P}' \) are directed acyclic graphs (DAG), where every argument is attacked by some other argument, then the conditions in Theorem 4 do indeed hold. So, for this class of naive DAS structural, static features do determine their dynamic properties.

**Corollary 5** Suppose that DAS \( \mathcal{P} \) and \( \mathcal{P}' \) are naive and statically bisimilar, and that \( \mathcal{A}_P \) and \( \mathcal{A}_P' \) are DAG where every argument is attacked by some other argument. Then, \( \mathcal{P} \) and \( \mathcal{P}' \) are dynamically bisimilar and therefore satisfy the same FO\(_A\)-ATL formulas.

### 4 Conclusions and Further Work

In many dialectical situations it is difficult to assume beforehand a bound on the number of arguments that agents have at their disposal. While some properties of such infinite argumentation frameworks have recently been investigated, their implication on debates amongst agents had not been analysed before. In this paper we set up and explored the framework of dynamic argumentation systems through a logic lens. We showed that in general static bisimilarity is not strong enough to capture equivalence of DAS (in terms of FO\(_A\)-ATL formulas they satisfy), and introduced a novel notion of dynamic bisimilarity. For some specific structures (in particular acyclic ones), the static notion remains powerful enough, at least for the naive agents studied here. While in this paper we have focused on such agents (unconstrained in their argumentative moves as long as they are relevant and truthful), an interesting direction of research is to investigate how more restricted protocols would also impact the dynamics of DAS. Another noteworthy feature of our framework is that formulas in FO\(_A\)-ATL can express the ability for a group of agents to...
ensure, for instance, that some arguments get accepted—thus paving the way for a logical analysis of multiparty protocols, see e.g. [Bonzon and Maudet, 2011].

Acknowledgements.
The third author benefited from the support of the project AMANDE ANR-13-BS02-0004 of the French National Research Agency (ANR).

References


