Formal Analysis of Dialogues on Infinite Argumentation Frameworks

F. Belardinelli[1] D. Grossi[2] N. Maudet[3]

[1]Laboratoire IBISC
Université d'Evry

[2]Department of Computer Science
University of Liverpool

[3]Sorbonne Universités
UPMC Univ Paris 06
CNRS, UMR 7606, LIP6
F-75005, Paris, France

Paris Dauphine – 17 June 2015

Outline

1. Motivation and Background:
   - the Dynamics of Argumentation

Outline

1. Motivation and Background:
   - the Dynamics of Argumentation

2. Main task: *formal* verification of *infinite-state* Dynamic Argumentation Systems (DAS)
   - model checking is appropriate for control-intensive applications...
     ...but less suited for data-intensive applications (data typically range over infinite domains) [1]

Outline

1. Motivation and Background:
   - the Dynamics of Argumentation
2. Main task: *formal* verification of *infinite-state* Dynamic Argumentation Systems (DAS)
   - model checking is appropriate for control-intensive applications...
     ...but less suited for data-intensive applications (data typically range over infinite domains) [1]
3. Key contributions:
   - DAS: a formal model for the dynamics of argumentation
   - FO-ATL: a specification language for DAS
   - truth preserving static and dynamic bisimulations

- The dialectical and dynamic dimensions of argumentation have been investigated since the inception of Dung's abstract argumentation theory [15, 16].
- However, the definition and analysis of 'static' justifiability criteria (i.e., argumentation semantics [2]) has come to form the backbone of abstract Argumentation Theory.
- Comparatively little work has been devoted to study forms of dynamic and multi-agent interaction.
  - Operationalizations of argumentation semantics via two-player games [19]
  - Analysis of strategic behavior in abstract forms of argumentation games [20, 22, 23]

We focus on the formal analysis of multi-agent strategic interactions (dialogues) on possibly infinite argumentation frameworks.

- agents are assumed to exchange arguments from possibly infinite AF
- they have private AF representing their 'views' on how arguments attack each other
- they interact by taking turns and attacking relevant arguments . . .
- . . . thus expanding the AF underlying the interaction

Claim: Dynamic Argumentation Systems (DAS) are general enough to model a wide range of dialogue protocols and games on abstract AF.

1. To specify (formally) dynamic properties of strategic interactions in argumentation
   - the proponent is able to respond to all attacks by maintaining a conflict-free set of arguments
   - the opponent has a strategy to force the proponent to run out of arguments
2. To develop techniques to tackle the verification problem (by model-checking)
   - how static/structural properties of argumentation frameworks influence their dynamic behavior?

Methodology: we capitalize on recent results on the verification of Data-aware Systems [7, 13, 18]

Model Checking in one slide

Model checking: technique(s) to **automatically** verify that a system design S satisfies a property P **before** deployment.

More formally, given

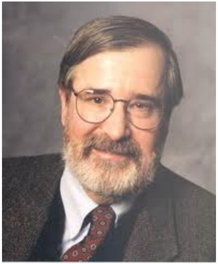
- a model $\mathcal{M}_S$ of system S
- a formula ϕ_P representing property P

we check that

$$\mathcal{M}_S \models \phi_P$$

Turing Award 2007

www.acm.org/press-room/news-releases-2008/turing-award-07



(a) E. Clarke (CMU, USA)

(b) A. Emerson (U. Texas, USA)

(c) J. Sifakis (IMAG, F)

- Jury justification

  "For their roles in developing model checking into a highly effective verification technology, widely adopted in the hardware and software industries."

Research questions

1. Which syntax and semantics to specify Dynamic Argumentation Systems?

Research questions

1. Which syntax and semantics to specify Dynamic Argumentation Systems?
2. Is verification of DAS decidable?

Research questions

1. Which syntax and semantics to specify Dynamic Argumentation Systems?
2. Is verification of DAS decidable?
3. If not, can we identify *relevant* fragments that are reasonably well-behaved?

Challenges

Multi-agent systems, but . . .

Challenges

Multi-agent systems, but . . .

- . . . states have a relational structure (argumentation frameworks),

Challenges

Multi-agent systems, but . . .

- . . . states have a relational structure (argumentation frameworks),
- arguments are potentially infinite,

Challenges

Multi-agent systems, but . . .

- . . . states have a relational structure (argumentation frameworks),
- arguments are potentially infinite,
- the state space is infinite in general.

Challenges

Multi-agent systems, but . . .

- . . . states have a relational structure (argumentation frameworks),
- arguments are potentially infinite,
- the state space is infinite in general.

$\Rightarrow$ The model checking problem cannot be tackled by standard techniques.

Related Work

- *Dynamics of argumentation*: how to change AF by performing operations on their structure? [5, 8, 9, 11, 14]
  - all references assume finite AF
- *Infinite Argumentation Frameworks*: infinite AF are gaining attention [3, 4, 6]
  - an infinity of arguments is critical in applications where upper bounds on the number of arguments cannot be established a priori
  - how to generalize known results for the finite case to the infinite case?
- *Logics for Abstract Argumentation*: several formalizations of argumentation theory have been put forward [12, 17]
  - languages sufficiently expressive to represent argumentation semantics
  - here the stress is on specifying the strategic abilities of agents engaging in a dialogue/dispute.

1. *Dynamic Argumentation Systems* (DAS) as a formal model.

1. *Dynamic Argumentation Systems* (DAS) as a formal model.
2. FO-ATL as a specification language:

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

opponent o can force proponent p to run out of moves in the next state.

1. *Dynamic Argumentation Systems* (DAS) as a formal model.
2. FO-ATL as a specification language:

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

   opponent o can force proponent p to run out of moves in the next state.
3. Bisimulation to tackle model checking.
   Main result: under specific conditions static features determine dynamic properties.

Basics: Argumentation Frameworks

Let $Ag = \{a_1, \ldots, a_n\}$ be a set of agent names.

Definition (Argumentation Framework)

A *(multi-agent) argumentation framework* is a tuple $\mathcal{A} = \langle A, \{\leftarrow_a\}_{a \in Ag} \rangle$ s.t.

- A is a (possibly infinite) set of arguments
- for every agent $a \in Ag$, $\leftarrow_a \subseteq A^2$ is an attack relation between arguments.

- We allow AF that include infinitely many arguments.
- $\mathcal{F}(A, Ag)$ is the set of all AF on sets A of arguments and Ag of agent names.

Language: First-order Logic

- Arguments call for First-order Logic.

  The specification language FO:

  $$\varphi \quad ::= \quad P(x) \mid \neg\varphi \mid \varphi \to \varphi \mid \forall y (A_a(y,x) \to \varphi[y]) \mid \forall y \varphi[y]$$

  where y is the only free variable in φ.

- The language FO is the dyadic fragment of first-order logic with one free variable.
  - equivalent to the multi-modal logic K with the universal modality [10].

Semantics: First-order Logic

Definition (IAF)

An *interpreted argumentation framework* is a couple $(\mathcal{A}, \pi)$ where

- π is an interpretation assigning a subset $\pi(P) \subseteq A$ to each predicate symbol P.

An argument $u \in A$ *satisfies* an FO-formula φ in an interpreted AF $(\mathcal{A}, \pi)$ iff

$(\mathcal{A}, \pi, u) \models P(x)$	iff	$u \in \pi(P)$
$(\mathcal{A}, \pi, u) \models \neg\psi$	iff	$(\mathcal{A}, \pi, u) \not\models \psi$
$(\mathcal{A}, \pi, u) \models \psi \to \psi'$	iff	$(\mathcal{A}, \pi, u) \not\models \psi$ or $(\mathcal{A}, \pi, u) \models \psi'$
$(\mathcal{A}, \pi, u) \models \forall y(A_a(y, x) \to \psi)$	iff	for every $v \in A$, $u \leftarrow_a v$ implies $(\mathcal{A}, \pi, v) \models \psi$
$(\mathcal{A}, \pi, u) \models \forall y\psi$	iff	for every $v \in A$, $(\mathcal{A}, \pi, v) \models \psi$

First-order Logic: Expressiveness

FO suffices to suffices to formalize several of the key notions from [16] (see also [17]).

$\pi(P)$ is conflict-free in $\mathcal{A}$	iff	$(\mathcal{A}, \pi) \models \forall x(P(x) \rightarrow \neg(\exists y(A(y,x) \wedge P(y)))$	$CFr(P)$
$\pi(P)$ is acceptable in $\mathcal{A}$	iff	$(\mathcal{A}, \pi) \models \forall x(P(x) \rightarrow \forall y(A(y,x) \rightarrow \exists z A(z,y) \wedge P(z)))$	$CFree(P)$
$\pi(P)$ is admissible in $\mathcal{A}$	iff	$\pi(P)$ is conflict-free and acceptable	$Adm(P)$
$\pi(P)$ is complete in $\mathcal{A}$	iff	$\pi(P)$ is conflict-free and $(\mathcal{A}, \pi) \models \forall x(P(x) \leftrightarrow \forall y(A(y,x) \rightarrow \exists z A(z,y) \wedge P(z)))$	$Cmp(P)$
$\pi(P)$ is a stable in $\mathcal{A}$	iff	$(\mathcal{A}, \pi) \models \forall x(P(x) \leftrightarrow \neg(\exists y(A(y,x) \wedge P(y)))$	$Stb(P)$

However, properties such as

- *a* belongs to the grounded extension
- *a* belongs to P, which is a preferred extension

are not expressible in FO.

To introduce interaction we start with a notion of agent.

Definition (Agent)

An *agent* is a tuple $a = \langle \mathcal{A}, Act, Pr \rangle$ where

- $\mathcal{A} \in \mathcal{F}(a)$ is the agent's *argumentation framework*
- the set Act contains actions
  - $attack(x, x')$, to attack argument x' with argument x
  - skip
- $Pr : \bigcup_{A' \subseteq A} \mathcal{F}(A', Ag) \mapsto 2^{Act(A)}$ is the *local protocol function*, where
  - for every $\mathcal{A}' \in \mathcal{F}(A', Ag)$, $attack(u, u') \in Pr(\mathcal{A}')$ only if $u' \in A'$ and $u' \leftarrow_a u$ holds in $\mathcal{A}$
  - the skip action is always enabled.

- The local state of agent a is modelled as an argumentation framework $\mathcal{A}$.
- By definition of protocol Pr, attacks must be *relevant* and *truthful* . . .

  . . . but they are not *compulsory*.

Example 1: Games for the Grounded Extensions

- Agents o and p hold the same private AF (i.e., $\mathcal{A}_o = \mathcal{A}_p$)
- for both agents we define the following protocol: if the current AF contains t_i then attack t_i with u_i or t_{i+1}, otherwise skip (i odd for opponent, i even for proponent)

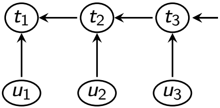


Figure : An infinite AF: each u_i and t_{i+1} attack each t_i.

Agents interact and generate DAS.

Definition (Global State)

Given a set Ag of agents $a_i = \langle \mathcal{A}_i, Act_i, Pr_i \rangle$ defined on the same (possibly infinite) set A of arguments, a *global state* is a couple (s, a) where

- $s \in \mathcal{F}(A', Ag)$ is an argumentation framework for some $A' \subseteq A$
- $a \in Ag$

- $\mathcal{G}$ is the set of all globales states.
- Some literature on agents and argumentation assumes that each agent is endowed with a distinct set of arguments (e.g., [21]).
- However, we can always consider the union of the sets of arguments for each agent.

Dynamic Argumentation System

We focus on dialogues between a proponent p and an opponent o.

Definition (DAS)

A *dynamic argumentation system* is a tuple $\mathcal{P} = \langle Ag, I, \tau, \pi \rangle$ where

- $Ag = \{o, p\}$
- $I \subseteq A \times \{o\}$ is the set of *initial global states* (s_0, o)
- $\tau : \mathcal{G} \times (Act_p(A) \cup Act_o(A)) \mapsto \mathcal{G}$ is the *transition function*, where
  1. $\tau((s, a), attack_{a'}(u, u'))$ is defined iff $a = a'$ and $attack_{a'}(u, u') \in Pr_{a'}(s)$
  2. $(s', a') = \tau((s, a), attack(u, u'))$ iff $a' \neq a$ and $s' = \langle A', \leftarrow' \rangle$ for $A' = A \cup \{u\}$ and $\leftarrow'_a = \leftarrow_a \cup \{(u', u)\}$
  3. $(s', a') = \tau((s, a), \text{skip})$ iff $a' \neq a$ and $s' = s$
- π 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 τ.
- DAS are infinite-state systems in general.
- DAS are first-order temporal structures.
  $\Rightarrow$ FO-ATL can be used as a specification language.

Example 2: Games for the Grounded Extensions

- the initial state is t_1
- the possible runs contain all sub-graphs of the AF generated from t_1

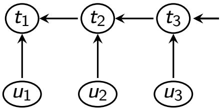


Figure : An infinite AF: each u_i and t_{i+1} attack each t_i.

Generated DAS

We consider the AF generated by a DAS.

Definition (Generated DAS)

Given a DAS $\mathcal{P}$ we define the corresponding (joint) AF $\mathcal{A}_{\mathcal{P}} = \langle A, \{\leftarrow_a\}_{a \in Ag} \rangle$ so that

- $u \leftarrow_a u'$ holds in $\mathcal{A}_{\mathcal{P}}$ iff $u \leftarrow_a u'$ holds in the AF $\mathcal{A}_a$ for agent $a \in Ag$.

Remark

Every reachable global state in $\mathcal{P}$ is a subgraph of $\mathcal{A}_{\mathcal{P}}$ $(*)$

- states in $\mathcal{P}$ are truthful, yet partial, representations of $\mathcal{A}_{\mathcal{P}}$
- the converse of $(*)$ does not hold in general, i.e., $\mathcal{P}$ needs not to include all subgraphs of $\mathcal{A}_{\mathcal{P}}$ as states
- this remark motivates the following definition

Generated DAS

Definition (Naive Agent)

An agent a is *naive* iff for every $\mathcal{A}' \in \mathcal{F}(A', Ag)$, $attack(u, u') \in Pr(\mathcal{A}')$ iff $u' \in A'$ and $u' \leftarrow_a u$ holds in $\mathcal{A}_a$.

An agent is *naive* if her protocol allows her to perform any available attack

Example

- the agents in the example above are naive
- therefore, we endow opponent o with a more restrictive protocol: if the current framework contains t_i then attack t_i with u_i, otherwise skip;
- this protocol makes o play more rationally, selecting arguments to which p cannot reply.

Specification Language: FO-ATL

- Arguments call for First-order Logic.
- Evolution calls for Temporal Logic.

The specification language FO-ATL:

$$\varphi \quad ::= \quad \psi \mid \neg\varphi \mid \varphi \to \varphi \mid \forall y(A_a(y,x) \to \varphi) \mid \forall y \varphi \mid \langle\langle N \rangle\rangle X\varphi \mid \langle\langle N \rangle\rangle G\varphi \mid \langle\langle N \rangle\rangle \varphi U \varphi$$

where $N \subseteq Ag$ and y is the only free variable in φ.

- An *N-strategy* is a mapping $f_N : \mathcal{S}^+ \mapsto \bigcup_{a \in N} Act_a(A)$ s.t. $f_N(\kappa \cdot (s,a)) \in Pr_a(s)$ for every $\kappa \in \mathcal{S}^+$.
- the *outcome* $out((s,a), f_N)$ of strategy f_N at state (s,a) is the set of all (s,a)-runs λ s.t. for every $b \in N$, $(\lambda(i+1), b') = \tau((\lambda(i), b), f_N(\lambda[0,i]))$ for all $i \geq 0$.

Definition (Semantics of FO-ATL)

An argument u *satisfies* a formula φ at state s in a DAS $\mathcal{P}$ iff

$(\mathcal{P}, s, u) \models \psi$ iff $(s, \pi, u) \models \psi$, if ψ is an FO-formula
$(\mathcal{P}, s, u) \models \langle\langle N \rangle\rangle X\varphi$ iff for some N-strategy f_N, for all $\lambda \in out(s, f_N)$, $(\mathcal{P}, \lambda(1), u) \models \varphi$
$(\mathcal{P}, s, u) \models \langle\langle N \rangle\rangle G\varphi$ iff for some N-strategy f_N, for all $\lambda \in out(s, f_N)$, $i \geq 0$, $(\mathcal{P}, \lambda(i), u) \models \varphi$
$(\mathcal{P}, s, u) \models \langle\langle N \rangle\rangle \varphi U \varphi'$ iff for some N-strategy f_N, for all $\lambda \in out(s, f_N)$, for some $k \geq 0$, $(\mathcal{P}, \lambda(k), u) \models \varphi'$ and for all j, $0 \leq j < k$ implies $(\mathcal{P}, \lambda(j), u) \models \varphi$
$(\mathcal{P}, s, u) \models \forall y(A_a(y,x) \to \varphi)$ iff for every $v \in s$, $u \leftarrow_a v$ implies $(\mathcal{P}, s, v) \models \varphi$
$(\mathcal{P}, s, u) \models \forall y \varphi$ iff for every $v \in s$, $(\mathcal{P}, s, v) \models \varphi$

The Model Checking Problem

- opponent o can force proponent p to run out of moves in the next state:

$$\langle\langle o \rangle\rangle X \forall x \neg \exists y A_p(y, x) \tag{1}$$

  this formula is true at argument t_1 in the DAS in the example above.

- 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):

$$P(x) \wedge \langle\langle p \rangle\rangle G\ \chi(P) \tag{2}$$

  where $\chi \in \{Cfr, Acc, Adm, Cmp, Stb\}$.

Definition (Model Checking Problem)

Given a DAS $\mathcal{P}$ and an FO-ATL sentence φ, determine whether $\mathcal{P} \models \varphi$.

Problem: the infinite domain A of arguments may generate infinitely many states!

Investigated solution: can we derive the *dynamic properties* of DAS from their *static features*?

Static Bisimulation

- A notion of bisimulation can naturally be defined on AF [17].

Definition (Static Bisimulation)

Let $(\mathcal{A}, \pi) = \langle A, \{\leftarrow_a\}_{a \in Ag}, \pi \rangle$ and $(\mathcal{A}', \pi') = \langle A', \{\leftarrow'_a\}_{a \in Ag}, \pi' \rangle$ be interpreted AF defined on a set Ag of agents.
A *static bisimulation* is a relation $S \subseteq A \times A'$ s.t. for $u \in A$, $u' \in A'$, $S(u, u')$ implies

(i) for every predicate symbol P, $u \in \pi(P)$ iff $u' \in \pi'(P)$;

(ii) for every $v \in A$, if $u \leftarrow_a v$ then for some $v' \in A'$, $u' \leftarrow'_a v'$ and $S(v, v')$;

(iii) 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 and u' are *bisimilar* ($u \simeq u'$) iff $S(u, u')$ for some static bisimulation S.
- two interpreted AF $\mathcal{A}$ and $\mathcal{A}'$ are *statically bisimilar* ($\mathcal{A} \simeq \mathcal{A}'$) iff
  - for every $u \in A$, $u \simeq u'$ for some $u' \in A'$
  - for every $u' \in A'$, $u' \simeq u$ for some $u \in A$

Lemma

Given bisimilar interpreted AF $(\mathcal{A}, \pi)$ and $(\mathcal{A}', \pi')$, and bisimilar arguments $u \in A$ and $u' \in A'$, then for every FO-*formula* φ,

$$(\mathcal{A}, \pi, u) \models \varphi \quad \textit{iff} \quad (\mathcal{A}', \pi', u') \models \varphi$$

Dynamic Bisimulation

- We extend bisimulation to dynamics.

Definition (Dynamic Bisimulation)

Given DAS $\mathcal{P}$ and $\mathcal{P}'$, a *dynamic simulation* is a relation $R \subseteq \mathcal{S} \times \mathcal{S}'$ s.t. for $s \in \mathcal{S}$, $s' \in \mathcal{S}'$, $R(s, s')$ implies:

1. $s \simeq s'$ for some static bisimulation S
2. for every $t \in \mathcal{S}$, if $s \rightarrow_a t$ then for some $t' \in \mathcal{S}'$, $s' \rightarrow'_a t'$, $t \simeq t'$ for some bisimulation $S' \supseteq S$, and $R(t, t')$.

A relation $D \subseteq \mathcal{S} \times \mathcal{S}'$ is a *dynamic bisimulation* iff both D and $D^{-1} = \{\langle s', s\rangle \mid D(s, s')\}$ are dynamic simulations.

$$\begin{tikzcd} \boxed{s} \arrow[r] \arrow[d, phantom, "\simeq"] & \boxed{t} \\ \boxed{s'} & \end{tikzcd}$$

Dynamic Bisimulation

- We extend bisimulation to dynamics.

Definition (Dynamic Bisimulation)

Given DAS $\mathcal{P}$ and $\mathcal{P}'$, a *dynamic simulation* is a relation $R \subseteq \mathcal{S} \times \mathcal{S}'$ s.t. for $s \in \mathcal{S}$, $s' \in \mathcal{S}'$, $R(s, s')$ implies:

1. $s \simeq s'$ for some static bisimulation S
2. for every $t \in \mathcal{S}$, if $s \rightarrow_a t$ then for some $t' \in \mathcal{S}'$, $s' \rightarrow'_a t'$, $t \simeq t'$ for some bisimulation $S' \supseteq S$, and $R(t, t')$.

A relation $D \subseteq \mathcal{S} \times \mathcal{S}'$ is a *dynamic bisimulation* iff both D and $D^{-1} = \{\langle s', s\rangle \mid D(s, s')\}$ are dynamic simulations.

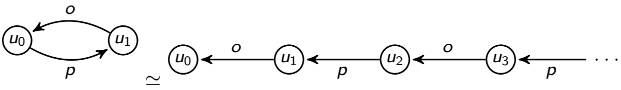
$$\begin{tikzcd}
\boxed{s} \arrow[r] \arrow[d, phantom, "\simeq"] & \boxed{t} \arrow[d, phantom, "\simeq"] \\
\boxed{s'} \arrow[r] & \boxed{t'}
\end{tikzcd}$$

- two states s and s' are *bisimilar* ($s \approx s'$) iff $D(u, u')$ for some dynamic bisimulation D.
- two DAS $\mathcal{P}$ and $\mathcal{P}'$ are *dynamically bisimilar* ($\mathcal{P} \approx \mathcal{P}'$) iff
  - for every initial state $s_0 \in \mathcal{P}$, $s_0 \approx s'_0$ for some $s'_0 \in \mathcal{P}'$
  - for every $s'_0 \in \mathcal{P}'$, $s_0 \approx s'_0$ for some $s_0 \in \mathcal{P}$
- two DAS $\mathcal{P}$ and $\mathcal{P}'$ are *statically bisimilar* iff $\mathcal{A}_{\mathcal{P}}$ and $\mathcal{A}_{\mathcal{P}'}$ are.

Static and Dynamic Bisimulation

Remark

Static bisimilarity does not imply dynamic bisimilarity, that is, there exist naive, statically bisimilar DAS $\mathcal{P}$ and $\mathcal{P}'$ such that $\mathcal{P} \not\approx \mathcal{P}'$.



(a) the AF $\mathcal{A}_{\mathcal{P}}$ and $\mathcal{A}_{\mathcal{P}'}$ are statically bisimilar.

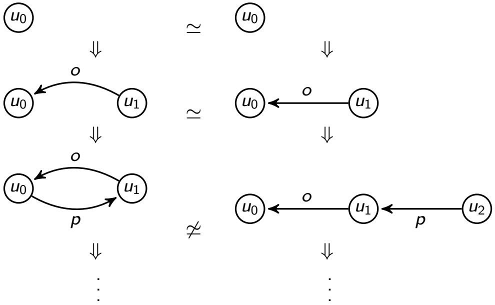


Figure : the DAS $\mathcal{P}$ and $\mathcal{P}'$ are statically bisimilar, but not dynamically bisimilar.

Preservation Result

Dynamically bisimilar DAS preserve the interpretation of FO-ATLformulas.

Theorem

Suppose that $s \approx s'$, and $u \simeq u'$ w.r.t. s and s'. Then for every FO-ATL *formula* φ*,*

$$(\mathcal{P}, s, u) \models \varphi \quad \textit{iff} \quad (\mathcal{P}', s', u') \models \varphi$$

From Static Properties to Dynamics

We can apply the result above to derive dynamic properties of DAS from their static features.

Theorem

Let $\mathcal{P}$ and $\mathcal{P}'$ be DAS. Suppose that $\mathcal{P}'$ is naive and for every $u \in s \in \mathcal{S}$, $u' \in s' \in \mathcal{S}'$, if $s \simeq s'$, $u \simeq u'$ w.r.t. s and s', and $u \leftarrow_a v$ in $\mathcal{A}_{\mathcal{P}}$ for some $v \in A$, then $u' \leftarrow'_a v'$ in $\mathcal{A}_{\mathcal{P}'}$ for some $v' \in A'$ 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, $D = \{(s, s') \mid s \simeq s'\}$ is a dynamic simulation between $\mathcal{P}$ and $\mathcal{P}'$.

Corollary

Suppose that DAS $\mathcal{P}$ and $\mathcal{P}'$ are naive and statically bisimilar, and that $\mathcal{A}_{\mathcal{P}}$ and $\mathcal{A}_{\mathcal{P}'}$ 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-ATL *formulas.*

Results

and main limitations

- Dynamic Argumentation Systems: a formal model for dialogues/disputes in AT
- The Specification Language FO-ATL
- Static and Dynamic Bisimulations for DAS
- Under specific conditions the static properties of DAS entail their dynamics

Next Steps

- Can we abstract a concrete, infinite-state DAS into a finite-state bisimilar DAS?
- If not, can we abstract the corresponding AF and then tranfer the result?
- What other dynamic properties of DAS can be derived from structural features?
- How can we develop efficient verification methods and techniques for DAS?

Thank you!

References

Christel Baier and Joost-Pieter Katoen.
Principles of Model Checking (Representation and Mind Series).
The MIT Press, 2008.

P. Baroni, M. Caminada, and M. Giacomin.
An Introduction to Argumentation Semantics.
The Knowledge Engineering Review, 26(4):365–410, 2011.

P. Baroni, F. Cerutti, P. E. Dunne, and M. Giacomin.
Computing with Infinite Argumentation Frameworks: The Case of AFRAs.
In *Theory and Applications of Formal Argumentation*, pages 197–214, 2012.

P. Baroni, F. Cerutti, P. E. Dunne, and M. Giacomin.
Automata for Infinite Argumentation Structures.
Journal of Artificial Intelligence, 203:104–150, 2013.

R. Baumann.
What does it take to enforce an argument? Minimal Change in Abstract Argumentation.
In *Proceedings of ECAI'12*, pages 127–132, 2012.

R. Baumann and C. Spanring.
Infinite Argumentation Frameworks.
In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*, pages 281–295, 2015.

F. Belardinelli, F. Patrizi, and A. Lomuscio.
Verification of Agent-Based Artifact Systems.
Journal of Artificial Intelligence Research, 51:333–77, 2014.

P. Besnard and S. Doutre.
Checking the Acceptability of a Set of Arguments.
In *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR'04)*, pages 59–64, 2004.