Abstract Agent Argumentation (Triple-A)*

Ryuta Arisaka[1], Ken Satoh[1], and Leendert van der Torre[2]

[1] National Institute of Informatics, Tokyo, Japan
[2] University of Luxembourg, Luxembourg

Abstract. In this paper we introduce a Dung style theory of abstract argumentation, which we call triple-A, in which each agent decides autonomously whether to accept or reject his own arguments. The agents may take some of the arguments of other agents into account, which we call their trusted arguments. An agent is called selfish if he ignores all arguments of other agents, and he is called social if he treats all arguments of other agents like his own. The extensions of globally accepted arguments are defined using a game theoretic equilibrium definition.

1 Introduction

In this paper we introduce triple-A, which stands for Abstract Agent Argumentation. In triple-A, we are interested in on the one hand individual agent acceptance, but on the other hand also global acceptance of arguments. In other words, we want to understand how autonomous agent acceptance leads to multi-agent interaction between arguing agents. Since the arguments an agent accepts *may* depend on the arguments the other agents accept, a formalisation of such interaction has to depend on game theoretic equilibria. In particular, in this paper we address the following questions:

1. How to define a dynamic variant of Dung's semantics that we can use for agent argumentation?
2. How to introduce agents in abstract argumentation, such that overall argument acceptance is an equilibrium between the individual agent acceptance functions?

To answer the first question, we define a semantics for abstract argumentation, which is based on selecting sub-frameworks from an argumentation framework. We call this a *dynamic* semantics, for reasons explained below. The Dung extensions can be retrieved from these sub-frameworks.

To answer the second question, in triple-A, an agent is an entity that *can* decide whether to accept or reject his own arguments independently of the arguments accepted by other agents, and we call such agents *acceptance autonomous*.

* The third author has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 690974 for the project MIREL: MIning and REasoning with Legal texts.

Examples are human agents, organisations, political parties, but increasingly also artificial agents.

We associate with every agent the set of arguments of other agents he trusts. Moreover, we assume that the agent takes the trusted arguments into account when deciding which arguments to accept. In this theory, an agent is called selfish if he ignores all arguments of other agents, and he is called social if he treats all arguments of other agents like his own. In this paper, *acceptance autonomy* is the only agent characteristic the triple-A theory is concerned with. As an application in the legal context, we cite modelling of a judicial panel system which involves several judges. Each judge may have his/her own argumentation process autonomously, but in interaction with other judges' argumentation, he/she may choose to adjust his/hers argumentation process.

We do not formally analyse the new theory yet using propositions and theorems, which is left to further research. This paper is based on discussions during a MIREL secondment of the third author of this paper to the NII institute. It combines former research of the authors, in particular the work on coalitional argumentation of the first two authors [1] and the work on multi-sorted and input/output argumentation of the third author [6, 2].

The layout of this paper is as follows. In Section 2 we introduce a dynamic variant of Dung semantics, which we need for agents taking the arguments of other agents into account. In Section 3 we introduce the triple-A theory. In Section 4 we illustrate it on some examples.

2 Abstract argumentation semantics

In this section we consider abstract argumentation semantics, and in the next section we introduce agents in this semantics. We first remind Dung's abstract argumentation semantics, which we call here a static semantics, then we introduce our dynamic semantics based on the selection of sub-frameworks, and finally we generalise to three valued labelings.

2.1 Static semantics

For completeness and reference below we briefly summarise Dung's abstract argumentation semantics. We say that a set $B \subseteq \mathcal{A}$ is *admissible*, if and only if it is *conflict-free* and it can *defend* each argument within the set. A set $B \subseteq \mathcal{A}$ is *conflict-free* if and only if there exist no arguments a_1 and a_2 in B such that $(a_1, a_2) \in \mathcal{R}$. Argument $a \in \mathcal{A}$ is *defended* by a set $B \subseteq \mathcal{A}$ (also called a_1 is *acceptable* with respect to B) if and only if for all $a_2 \in \mathcal{A}$, if $(a_2, a_1) \in \mathcal{R}$, then there exists $a_3 \in B$ such that $(a_3, a_2) \in \mathcal{R}$. Based on the notion of admissible sets, Dung defines various kinds of extensions. Formally, we have the following definition.

Definition 1 (Dung semantics). *Let $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ be a graph called an argumentation framework, and $B \subseteq \mathcal{A}$ a set of arguments.*

- B *is* conflict-free *if and only if* $\nexists a_1, a_2 \in B$, *s.t.* $(a_1, a_2) \in \mathcal{R}$.
- *An argument* $a_1 \in \mathcal{A}$ *is defended by* B *(equivalently* a_1 *is* acceptable *w.r.t.* B*), if and only if* $\forall (a_2, a_1) \in \mathcal{R}$, $\exists \gamma \in B$, *s.t.* $(a_3, a_2) \in \mathcal{R}$.
- B *is* admissible *if and only if* B *is conflict-free, and each argument in* B *is defended by* B.
- B *is a* complete *extension if and only if* B *is admissible and each argument in* $\mathcal{A}$ *that is defended by* B *is in* B.
- B *is a* preferred *extension if and only if* B *is a maximal (w.r.t. set-inclusion) complete extension.*
- B *is a* grounded *extension if and only if* B *is the minimal (w.r.t. set-inclusion) complete extension.*
- B *is a* stable *extension if and only if* B *is conflict-free, and* $\forall a_1 \in \mathcal{A} \setminus B$, $\exists a_2 \in B$ *s.t.* $(a_2, a_1) \in \mathcal{R}$.

We use $sem \in \{cmp, prf, grd, stb\}$ to denote complete, preferred, grounded, or stable semantics, respectively. A set of argument extensions of $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ is denoted as $sem(\mathcal{F})$.

Finally, we note that the grounded extension is special for several reasons, namely: it is unique, it is contained in all other Dung extensions, and it can be defined as fixpoint of the so-called characteristic function (mapping sets of arguments to sets of arguments they defend) applied to the empty set. Grounded semantics also plays a distinguished role in the dynamic Dung semantics below.

2.2 Labelling semantics

In this paper we use also the labelling-based approach to the definition of argumentation semantics. It is well known, see e.g. Baroni *et al.*'s overview [3], that for the semantics considered in this paper there is a direct correspondence with the 'traditional' extension-based approach.

A labelling assigns to each argument of an argumentation framework a label taken from a predefined set Λ. For technical reasons, we define labellings both for argumentation frameworks and for arbitrary sets of arguments.

Definition 2. *Let* $\Lambda = \{\texttt{in}, \texttt{out}, \texttt{undec}\}$ *be a set of labels. Given a set of arguments* B*, a* labelling *of* B *is a total function* $Lab : B \longrightarrow \Lambda$*. The set of all* labellings *of* B *is denoted as* $\mathfrak{L}_B$*. Given an argumentation framework* $\mathcal{F} = (\mathcal{A}, \mathcal{R})$*, a* labelling *of* $\mathcal{F}$ *is a labelling of* $\mathcal{A}$*. The set of all* labellings *of* $\mathcal{F}$ *is denoted as* $\mathfrak{L}(\mathcal{F})$*. For a labelling* Lab *of* B*, the restriction of* Lab *to a set of arguments* $B' \subseteq B$*, denoted as* $Lab{\downarrow_{B'}}$*, is defined as* $Lab \cap (B' \times \Lambda)$.

The label `in` means that the argument is accepted, the label `out` means that the argument is rejected, and the label `undec` means that the status of the argument is undecided. Given a labelling Lab, we write $\texttt{in}(Lab)$ for $\{a \mid Lab(a) = \texttt{in}\}$, $\texttt{out}(Lab)$ for $\{a \mid Lab(a) = \texttt{out}\}$ and $\texttt{undec}(Lab)$ for $\{a \mid Lab(a) = \texttt{undec}\}$.

A labelling-based semantics prescribes a set of labellings for each argumentation framework.

Definition 3. *Given an argumentation framework* $\mathcal{F} = (\mathcal{A}, \mathcal{R})$*, a labelling-based semantics* $\mathbf{S}$ *associates with* $\mathcal{F}$ *a subset of* $\mathfrak{L}(\mathcal{F})$*, denoted as* $\mathbf{L_S}(\mathcal{F})$.

2.3 Baroni *et al.*'s notion of local function

In this section we repeat some basic concepts regarding local functions from Baroni *et al.*, we refer to their paper [2] for further explanations and examples.

Intuitively, given an argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ and a subset B of its arguments, the elements affecting $\mathcal{F}{\downarrow_B}$, which is $(\{a \in \mathcal{A} \mid a \in B\}, \{(a_1, a_2) \in \mathcal{R} \mid a_1, a_2 \in B\})$, include the arguments attacking B from the outside, called *input* arguments, and the attack relation from the input arguments to B, called *conditioning relation*.

Definition 4 (Input [2]). *Given $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ and a set $B \subseteq \mathcal{A}$, the* input *of B, denoted as B^{inp}, is the set $\{a_2 \in \mathcal{A} \setminus B \mid \exists a_1 \in B, (a_2, a_1) \in \mathcal{R}\}$, the* conditioning relation *of B, denoted as B^R, is defined as $\mathcal{R} \cap (B^{inp} \times B)$.*

An *argumentation framework with input* consists of an argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ (playing the role of a partial argumentation framework), a set of external input arguments $\mathcal{I}$, a labelling $L_\mathcal{I}$ assigned to them and an attack relation $R_\mathcal{I}$ from $\mathcal{I}$ to $\mathcal{A}$. A *local function* which, given an argumentation framework with input, returns a corresponding set of labellings of $\mathcal{F}$.

Definition 5 (Framework with input [2]). *An* argumentation framework with input *is a tuple $(\mathcal{F}, \mathcal{I}, L_\mathcal{I}, R_\mathcal{I})$, including an argumentation framework $\mathcal{F} = (\mathcal{A}, \mathcal{R})$, a set of arguments $\mathcal{I}$ such that $\mathcal{I} \cap \mathcal{A} = \emptyset$, a labelling $L_\mathcal{I} \in \mathfrak{L}_\mathcal{I}$ and a relation $R_\mathcal{I} \subseteq \mathcal{I} \times \mathcal{A}$. A* local function *assigns to any argumentation framework with input a (possibly empty) set of labellings of $\mathcal{F}$, i.e. $f(F, \mathcal{I}, L_\mathcal{I}, R_\mathcal{I}) \in 2^{\mathfrak{L}(F)}$.*

A similar notion appears also in [5, 4].

For any semantics, a "sensible" local function, called *canonical local function*, is the one that describes the labellings of the so-called standard argumentation frameworks.

Definition 6 (Standard argumentation framework [2]). *Given an argumentation framework with input $(\mathcal{F}, \mathcal{I}, L_\mathcal{I}, R_\mathcal{I})$, the standard argumentation framework w.r.t. $(\mathcal{F}, \mathcal{I}, L_\mathcal{I}, R_\mathcal{I})$ is defined as $\mathcal{F}' = (\mathcal{A} \cup \mathcal{I}', \mathcal{R} \cup R'_\mathcal{I})$, where $\mathcal{I}' = \mathcal{I} \cup \{a' \mid a \in \mathtt{out}(L_\mathcal{I})\}$ and $R'_\mathcal{I} = R_\mathcal{I} \cup \{(a', a) \mid a \in \mathtt{out}(L_\mathcal{I})\} \cup \{(a, a) \mid a \in \mathtt{undec}(L_\mathcal{I})\}$.*

Roughly, the standard argumentation framework puts $\mathcal{F}$ under the influence of $(\mathcal{I}, L_\mathcal{I}, R_\mathcal{I})$, by adding $\mathcal{I}$ to $\mathcal{A}$ and $R_\mathcal{I}$ to $\mathcal{R}$, and by enforcing the label $L_\mathcal{I}$ for the arguments of $\mathcal{I}$ in this way:

- for each argument $a \in \mathcal{I}$ such that $L_\mathcal{I}(a) = \mathtt{out}$, an unattacked argument a' is included which attacks a, in order to get A labelled `out` by all labellings of $\mathcal{F}'$;
- for each argument $a \in \mathcal{I}$ such that $L_\mathcal{I}(a) = \mathtt{undec}$, a self-attack is added to a in order to get it labelled `undec` by all labellings of F';
- each argument $a \in \mathcal{I}$ such that $L_\mathcal{I}(a) = \mathtt{in}$ is left unattacked, so that it is labelled `in` by all labellings of $\mathcal{F}'$.

Definition 7 (Canonical local function [2]).
Given a semantics $\mathbf{S}$*, the* canonical local function *of* $\mathbf{S}$ *(also called local function of* $\mathbf{S}$*) is defined as* $f_{\mathbf{S}}(\mathcal{F}, \mathcal{I}, L_{\mathcal{I}}, R_{\mathcal{I}}) = \{Lab{\downarrow_{\mathcal{A}}} \mid Lab \in \mathbf{L_S}(\mathcal{F}')\}$*, where* $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ *and* $\mathcal{F}'$ *is the standard argumentation framework w.r.t.* $(\mathcal{F}, \mathcal{I}, L_{\mathcal{I}}, R_{\mathcal{I}})$*.*

2.4 Dynamic semantics

In this section we introduce a new dynamic semantics for abstract argumentation. It is based on the intuition that agents do not either accept or reject all their arguments when they interact with other agents. In particular, arguments they do not fully accept when considering their own arguments only, may be accepted when also the arguments of other agents are considered. Dynamic semantics is a generalisation of static Dung semantics.

A dynamic semantics associates sub-frameworks with a framework. This can be given a dynamic interpretation as follows. Consider paths of sub-frameworks between an argumentation framework and its extensions. Note that an extension itself is also a sub-framework of the original framework. At each step of a framework-extension path, some attacks and arguments are removed from the framework. When attacks are removed, it typically means that conflicts are resolved by breaking cycles. When arguments are left out (together with the attacks they are involved in) it typically means that the argument is no longer considered for any of the further refinements of the framework.

To bridge dynamic and static semantics, we use the grounded extension. In particular, we use the grounded semantics to associate with each sub-framework a unique extension. This can also be interpreted dynamically. At each step of the sequence, including the initial and the final step, we consider the Dung grounded extension as the intermediate meaning of the sub-framework. Monotonic sequences are sequences of sub-frameworks of which the grounded extension is monotonically increasing (or stays constant).

The notions of static and dynamic semantics and their relation are formalised in the following general definition.

Definition 8 (Static and dynamic semantics). *We say that* $\mathcal{F}' = (\mathcal{A}', \mathcal{R}')$ *is a sub-framework of* $\mathcal{F} = (\mathcal{A}, \mathcal{R})$*, written as* $\mathcal{F}' \subseteq \mathcal{F}$*, if and only if* $\mathcal{A}' \subseteq \mathcal{A}$ *and* $\mathcal{R}' \subseteq \mathcal{R}$*.*

A static semantics sem is a function from argumentation frameworks to sets of their arguments, i.e. if $B \in sem((\mathcal{A}, \mathcal{R}))$ *then* $B \subseteq \mathcal{A}$*.*

A dynamic semantics S is a function from argumentation frameworks to sets of their sub-frameworks, i.e. if $\mathcal{F}' \in S(\mathcal{F})$ *then* $\mathcal{F}' \subseteq \mathcal{F}$*.*

The static semantics sem implied by dynamic semantics S is $sem(\mathcal{F}) = \cup\{grd(\mathcal{F}') \mid \mathcal{F}' \in S(\mathcal{F})\}$,[3] *i.e. the grounded extensions of* $S(\mathcal{F})$*.*

[3] Note that this equation contains a union symbol $\cup$, because semantics are defined as sets of extensions. Since grounded semantics contains only a singe extension it could have been defines as a function from frameworks to a single set of arguments, i.e. $grd((\mathcal{A}, \mathcal{R})) \subseteq \mathcal{A}$, and then we would have $sem(\mathcal{F}) = \{grd(\mathcal{F}') \mid \mathcal{F}' \in S(\mathcal{F})\}$.

We now consider the dynamics. For example, consider an argument framework $\mathcal{F}$ where argument a attacks argument b. The dynamic semantics can be $\mathcal{F}$ or the single argument framework containing a only. Both express the static extension a. However, in the first case, if we later remove argument a, then we accept argument b. In the second case, if we remove a, then we do not accept any argument. We call the former *open-minded* and the latter *stubborn*. More specifically:

Definition 9 (Open-minded and stubborn semantics). *We write $\mathcal{F}_1 \prec_{sem} \mathcal{F}_2$ just when $\mathcal{F}_1$ is a sub-framework of $\mathcal{F}_2$ and they both have the same static semantics sem. We say that $S(\mathcal{F})$ is open-minded just when each $\mathcal{F}' \in S(\mathcal{F})$ is such that either $\mathcal{F}'' \prec_{grd} \mathcal{F}'$ or else $\mathcal{F}''$ and $\mathcal{F}'$ are not comparable in $\prec_{grd}$ for each sub-framework $\mathcal{F}''$ of $\mathcal{F}$. We say that $S(\mathcal{F})$ is stubborn just when each $\mathcal{F}' \in S(\mathcal{F})$ is such that either $\mathcal{F}' \prec_{grd} \mathcal{F}''$ or else $\mathcal{F}''$ and $\mathcal{F}'$ are not comparable in $\prec_{grd}$ for each sub-framework $\mathcal{F}''$ of $\mathcal{F}$.*

For a later reference, we say that $\mathcal{F}_1$ is *sem*-maximal in $\mathcal{F}$ just when we either have $\mathcal{F}_2 \prec_{sem} \mathcal{F}_1$ or else $\mathcal{F}_2$ and $\mathcal{F}_1$ are not comparable in $\prec_{sem}$ for each sub-framework $\mathcal{F}_2$ of $\mathcal{F}$.

While the comparison of the two variations may be of interest, in the remainder of this paper, we consider only open-minded semantics.

We now consider dynamic generalisations of static Dung semantics. It works as follows. Assume a given static Dung semantics. Now a dynamic semantics for this static semantics satisfies the natural criterion that extensions of the framework coincide with the grounded extensions of the sub-frameworks.

Definition 10 (Dynamic Dung semantics). *Consider a Dung semantics sem. A dynamic semantics S is a dynamic generalisation of sem iff the following holds:*

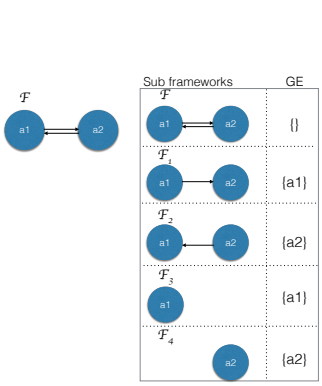
$$\cup\{grd(\mathcal{F}') \mid \mathcal{F}' \in S(\mathcal{F})\} = sem(\mathcal{F})$$

By $S_\alpha(\mathcal{F})(\alpha \in \{cmp, prf, grd, stb\})$, we denote a dynamic semantics that is a dynamic generalisation of α.

Dynamic Dung semantics is illustrated in Example 1 below.

Example 1.
Consider the argumentation framework $\mathcal{F} = (\{a_1, a_2\}, \{(a_1, a_2), (a_2, a_1)\})$ in Figure 1. It has five sub-frameworks: $\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$, and $\mathcal{F}_4$ with a corresponding grounded extension $\{\}, \{a_1\}, \{a_2\}, \{a_1\}$ and $\{a_2\}$. For $\mathcal{F}$, we have 9 dynamic generalisations of *cmp*, prf and *stb*, and one dynamic generalisation, $\mathcal{F}$ itself, of *grd*, as the tables show. However, as $\mathcal{F}_3 \prec_{grd} \mathcal{F}_1$ and $\mathcal{F}_4 \prec_{grd} F_2$, there is only one open-minded semantics for each of *cmp* ($\{\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2\}$), prf, stb ($\{\mathcal{F}_1, \mathcal{F}_2\}$) and *grd* ($\{\mathcal{F}\}$). (Just to note, there is only one stubborn dynamic semantics again for each of *cmp* ($\{\mathcal{F}, \mathcal{F}_3, \mathcal{F}_4\}$), prf, stb ($\{\mathcal{F}_3, \mathcal{F}_4\}$) and *grd* ($\{\mathcal{F}\}$).)

There is much more to be said abut dynamic semantics and its relation to other argumentation semantics. However, since we are primarily interested in



Static Semantics

cmp	prf	grd	stb
{{},{a1},{a2}}	{{a1},{a2}}	{}	{{a1},{a2}}

Dynamic Variants

cmp	prf	grd	stb
$\{\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2\}$,	$\{\mathcal{F}_1, \mathcal{F}_2\}$,	$\{\mathcal{F}\}$	$\{\mathcal{F}_1, \mathcal{F}_2\}$,
$\{\mathcal{F}, \mathcal{F}_1, \mathcal{F}_4\}$,	$\{\mathcal{F}_1, \mathcal{F}_4\}$,		$\{\mathcal{F}_1, \mathcal{F}_4\}$,
$\{\mathcal{F}, \mathcal{F}_3, \mathcal{F}_2\}$,	$\{\mathcal{F}_3, \mathcal{F}_2\}$,		$\{\mathcal{F}_3, \mathcal{F}_2\}$,
$\{\mathcal{F}, \mathcal{F}_3, \mathcal{F}_4\}$,	$\{\mathcal{F}_3, \mathcal{F}_4\}$,		$\{\mathcal{F}_3, \mathcal{F}_4\}$,
$\{\mathcal{F}, \mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_2\}$,	$\{\mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_2\}$,		$\{\mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_2\}$,
$\{\mathcal{F}, \mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_4\}$,	$\{\mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_4\}$,		$\{\mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_4\}$,
$\{\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_4\}$,	$\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_4\}$,		$\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_4\}$,
$\{\mathcal{F}, \mathcal{F}_3, \mathcal{F}_2, \mathcal{F}_4\}$,	$\{\mathcal{F}_3, \mathcal{F}_2, \mathcal{F}_4\}$,		$\{\mathcal{F}_3, \mathcal{F}_2, \mathcal{F}_4\}$,
$\{\mathcal{F}, \mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_2, \mathcal{F}_4\}$	$\{\mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_2, \mathcal{F}_4\}$		$\{\mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_2, \mathcal{F}_4\}$

Fig. 1. The left sub-figure shows an argumentation framework $\mathcal{F}$, its sub-frameworks and their grounded extensions. The right sub-figure shows static semantics and corresponding dynamic generalisations.

agent argumentation in this paper, we leave some further observations to future work, and we continue with the extension of dynamic semantics with agents.

3 Triple-A theory

We consider a AAA framework as a Dung framework with an equivalence relation on arguments, which represents a partitioning of the arguments over what we call agents. To facilitate the upcoming definitions, we represent the set of agents explicitly by the pair (Ag, Src) in the AAA framework, rather than by a more abstract equivalence relation.

Definition 11 (AAA framework). *An abstract agent argumentation (or AAA) framework is a tuple $\mathcal{F} = (\mathcal{A}, \mathcal{R}, Ag, Src)$ where $\mathcal{A}$ is a non-empty set (of arguments), $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation over arguments (expressing attack), Ag is a set (of agents) and $Src : \mathcal{A} \to Ag$ is a function mapping each argument to the agent that put it forward (also known as its source).*

We define input and trust for sets of agents (following conventions in game theory we call these sets of agents *coalitions*), though in some cases we may primarily be interested in the case where the set consists of a single agent.

Definition 12 (Trust). *For AAA framework $\mathcal{F} = \langle \mathcal{A}, \mathcal{R}, Ag, Src \rangle$ and coalition $C \subseteq Ag$, the agent trust is a function $T(\mathcal{F}, C) \subseteq \mathcal{A}$ that maps coalition $C \subseteq Ag$ to a set of arguments, representing the arguments the coalition trusts. A coalition is called social if $T(\mathcal{F}, C) = \mathcal{A}$ and selfish if $T(\mathcal{F}, C) = \{a \in \mathcal{A} \mid Src(a) = C\}$.*

The input of C, written as I_C, is the set of arguments $\{a_1 \in T(\mathcal{F}, C) \mid Src(a_1) \neq C, \exists a_2 \in Arg : Src(a_2) = C, Att(a_1, a_2)\}$.

Agent semantics is a local function with respect to the agent input. With respect to the definitions of Baroni et al, the only difference is that the local function maps a framework with input to a set of sub-frameworks.

Definition 13 (Local agent semantics). *Given a semantics S, the agent semantics is a local function of S, defined as $f_S(\mathcal{F}, I_C, L_{I_C}, R_{I_C}) = \{\mathcal{F}''{\downarrow_{\mathcal{A}}} \mid \mathcal{F}'' \in S(\mathcal{F}')\}$, where $\mathcal{F} = (\mathcal{A}, \mathcal{R})$ and F' is the standard argumentation framework w.r.t. $(\mathcal{F}, I_C, L_{I_C}, R_{I_C})$.*

We are finally ready to give the central definition of our multi-agent interaction. Since the local semantics is expressed as sub-frameworks, but the input is expressed as a labelling, we represent the equilibria by pairs of an argumentation framework and a labelling.

Definition 14 (Multi-agent semantics). *Let a coalition structure CS be an equivalence relation over agents, expressing a partitioning of the agents into coalitions $(C_1, \ldots, C_n)$.*

The semantics of a triple-A framework $\mathcal{F}$ with coalition structure CS is a pair of an argumentation framework $\mathcal{F}'$ and a labelling L such that:

1. *L is a labelling of $\mathcal{F}$ that coincides with the labelling of $\mathcal{F}'$ for the arguments in $\mathcal{F}'$ and gives label out to all other arguments,*
2. *For each $C \in CS$, $\mathcal{F}'$ replaces $\mathcal{F}_C$ by $\mathcal{F}'_C \in f_S(\mathcal{F}_C, I_C, L_{I_C}, R_{I_C})$.*

4 Examples

In this section, we illustrate AAA theory through some examples.

Example 2. Figure 2 illustrates differences between social and selfish agents (which we call coalitions, as we stated earlier), and how they may influence multi-agent semantics. In the legal context, these agents may be judges participating in a judiciary panel, as we mentioned in Section 1. In the argumentation framework $\mathcal{F}$ as given in (A), C_1 and C_2 can be social or selfish. Let us consider preferred local agent semantics. When C_1 is selfish, all attacks from external arguments (just a_3 in this example) on its members are ignored, so $I_{C_1} = \emptyset$ (see (C)). According to Definition 10 then, we have 9 dynamic generalisations of static preferred semantics $\{\{a_1\}, \{a_2\}\}$ for the local argumentation framework of selfish C_1 (see Example 1). Of these, $\{\mathcal{F}_{(1a)}, \mathcal{F}_{(1b)}\}$ $(= f_{prf}(\mathcal{F}{\downarrow_{\{a_1,a_2\}}}, \emptyset, L_\emptyset, R_\emptyset))$ is open-minded dynamic preferred semantics of selfish C_1. To explain $f_{S_{prf}}(\mathcal{F}{\downarrow_{\{a_1,a_2\}}}, \emptyset, L_\emptyset, R_\emptyset)$, it is $\{\mathcal{F}{\downarrow_{\{a_1,a_2\}}} \mid \mathcal{F}'' \in S_{prf}(\mathcal{F}')\}$ where $\mathcal{F} = (\{a_1, a_2\}, \{(a_1, a_2), (a_2, a_1)\}) = \mathcal{F}_B$ by Definition 13 and $\mathcal{F}'$ is the standard argumentation framework with respect to $(\mathcal{F}, \emptyset, L_\emptyset, R_\emptyset)$ which is just $\mathcal{F}_B$. Hence in the open-minded preferred dynamic semantics we obtain *grd*-maximal sub-frameworks of $\mathcal{F}_B$ that correspond to the static preferred semantics of $\mathcal{F}_B$ $(\{\{a_1\}, \{a_2\}\})$. Meanwhile, the open-minded dynamic preferred semantics of C_2 is $\{a_3\}$ $(= f_{S_{prf}}(\mathcal{F}{\downarrow_{\{a_3\}}}, \emptyset, L_\emptyset, R_\emptyset))$, as it is not attacked by any arguments. Multi-agent semantics of $\mathcal{F}$ with (C_1, C_2) combines these sub-frameworks of C_1, of C_2, and any attacks between them, as

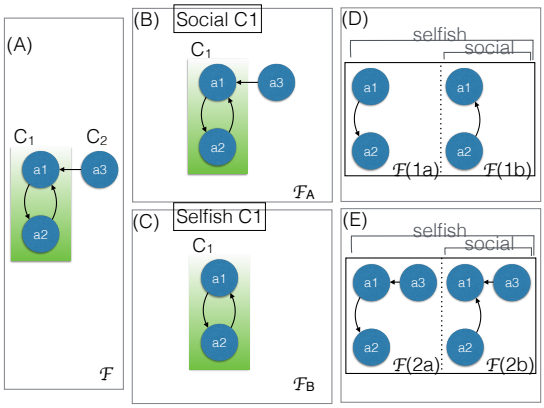


Fig. 2. (A) An argumentation framework $\mathcal{F}$ with 3 arguments and attacks. C_1, and C_2, comprises a_1 and a_2, and a_3. (B) Social C_1 with its input $I_{C_1} = \{a_3\}$. (C) Selfish C_1 with its input $I_{C_1} = \emptyset$. (D) Two (out of three) *grd*-maximal sub-frameworks of selfish C_1's local argumentation framework. $\mathcal{F}_{(1b)}$ is the only one *grd*-maximal sub-framework of social C_1's local argumentation framework. (E) Composition of: (1) the two sub-frameworks of C_1's local argumentation framework in (D); (2) the sub-framework of C_2's local argumentation framework which is $\{a_3\}$; and (3) the attack between them for multi-agent semantics.

shown in (E) in Figure 2. Consequently, for selfish C_1, we have the open-minded dynamic preferred semantics $\{\mathcal{F}_{(2a)}, \mathcal{F}_{(2b)}\}$ with a corresponding static preferred semantics $\{\{a_2, a_3\}\}$, i.e. the labels are: $\{\{a_1 : \mathsf{out}, a_2 : \mathsf{in}, a_3 : \mathsf{in}\}\}$.

When C_1 is on the other hand social, I_{C_1} is $\{a_3\}$, and we have $\{\mathcal{F}'_{(2b)}\}$ as C_1's open-minded dynamic preferred semantics with a corresponding static preferred semantics $\{\{a_2, a_3\}\}$. This is derived from $f_{S_{prf}}(\mathcal{F}{\downarrow_{\{a_1,a_2\}}}, \{a_3\}, L_{\{a_3\}}, R_{\{a_3\}})$, which, by Definition 13, is $\{\mathcal{F}{\downarrow_{\{a_1,a_2\}}} \mid \mathcal{F}'' \in S_{prf}(\mathcal{F}')\}$ where $\mathcal{F} = \mathcal{F}_B$ and $\mathcal{F}'$ is the standard argumentation framework with respect to $(\mathcal{F}, \{a_3\}, L_{a_3}, R_{a_3})$ which is $\mathcal{F}_A$.

Here, we would like to direct readers' attention to the case with selfish C_1. Whereas with Definition 7 we obtain $\{\{a_1 : \mathsf{in}, a_2 : \mathsf{out}\}, \{a_1 : \mathsf{out}, a_2 : \mathsf{in}\}\}$ for preferred semantics, and if, after obtaining the semantics, we add a_3 and its attack on a_1, we no longer remember the attacks in C_1, and the merged preferred labels we would obtain are $\{a_3 : \mathsf{in}, a_1 : \mathsf{out}, a_2 : \mathsf{out}\}$ and $\{a_3 : \mathsf{in}, a_1 : \mathsf{out}, a_2 : \mathsf{in}\}$, our open-minded dynamic semantics provides us with richer information - a rationale in a way - as to why a_1 or a_2 should be labelled in or out (or undec). For instance $\mathcal{F}'_{(1a)}$ retains, in addition to that a_1 is in, the information that a_2 is out *because* a_1 *is in*. It is because of this kind of additional information that the static semantics locally obtained for a coalition, which is the same as with Definition 7, can be adjusted later in a coalition structure, that is, the preferred local

agent semantics of C_1, in combination to the preferred local agent semantics of C_2, generates $\mathcal{F}_{(2b)}$ which has the static preferred extension of $\{\{a_2, a_3\}\}$, as expected from F. We believe our sub-framework-based local function adds further flexibility to the label-based local function [2].

This static semantics adjustment under open-minded dynamic semantics may be a matter of which arguments a coalition should consider certainly worthwhile to take into account. Let us suppose some coalition C_1. It may represent a judge with his/her arguments. Then we may reasonably assume that C_1 regards all his/her internal arguments worthy of attention. Hence it evaluates all of them for deciding which ones he/she should accept or reject. This explains why a AAA local function considers all arguments of C_1 whether or not C_1 is social. When C_1 and C_2 form a coalition structure, the structure then becomes a single entity with arguments, which could be a judicial panel of judges. We may reasonably assume that each individual judge in the judicial panel pays attention to the other judges. This explains why, under open-minded dynamic semantics, the static semantics of a local agent semantics put forward by a judge can dynamically change when it interacts with other judges in the judicial panel.

Also, an agent can trust arguments outside it. Agents trust is defined for

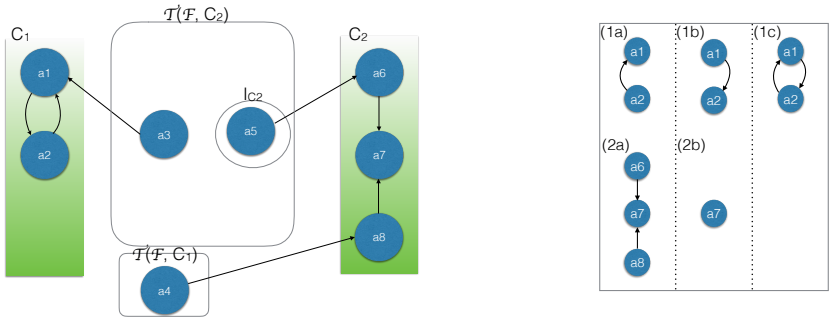


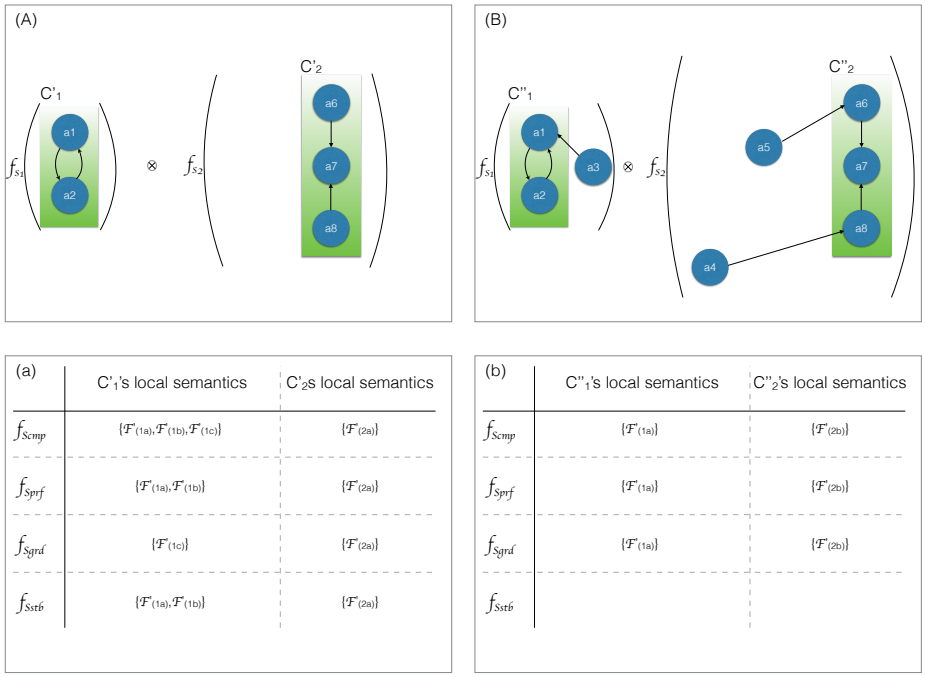
Fig. 3. An argumentation framework with two coalitions C_1 and C_2, and (open-minded) sub-frameworks: (1a) - (1c) for C_1 and (2a) - (2b) for C_2.

that purpose. If an argument is trusted by an agent, then the agent considers it certainly worthwhile to take into account. When the argument attacks the agent, it reflects the attack in deciding which internal arguments it accepts or rejects. While we have already encountered two extreme cases of a selfish or a soial agent, our trust functioin covers any other cases. Trust of a coalition structure can be based on the trust of agents in the structure. For example, we may have that a coalition structure trusts an argument if there is at least one agent in the coalition structure that trusts that argument. As the other extreme, a more

skeptical notion of coalitional trusts says that all constituting agents have to trust an argument, before the coalition structure trusts it.

Definition 15. *Credulous agent trust is agent trust satisfying* $T(\mathcal{F}, C_1 \cup C_2) = T(\mathcal{F}, C_1) \cup T(\mathcal{F}, C_2)$.

Skeptical agent trust is agent trust satisfying $T(\mathcal{F}, C_1 \cap C_2) = (T(\mathcal{F}, C_1) \cap T(\mathcal{F}, C_2)) \cup \{a \in \mathcal{A} \mid Src(a) = C_1 \text{ or } Src(a) = C_2\}$.



(a)

	C'1's local semantics	C'2s local semantics
f_{Scmp}	$\{\mathcal{F}'_{(1a)}, \mathcal{F}'_{(1b)}, \mathcal{F}'_{(1c)}\}$	$\{\mathcal{F}'_{(2a)}\}$
f_{Sprf}	$\{\mathcal{F}'_{(1a)}, \mathcal{F}'_{(1b)}\}$	$\{\mathcal{F}'_{(2a)}\}$
f_{Sgrd}	$\{\mathcal{F}'_{(1c)}\}$	$\{\mathcal{F}'_{(2a)}\}$
f_{Sstb}	$\{\mathcal{F}'_{(1a)}, \mathcal{F}'_{(1b)}\}$	$\{\mathcal{F}'_{(2a)}\}$

(b)

	C''1's local semantics	C''2's local semantics
f_{Scmp}	$\{\mathcal{F}'_{(1a)}\}$	$\{\mathcal{F}'_{(2b)}\}$
f_{Sprf}	$\{\mathcal{F}'_{(1a)}\}$	$\{\mathcal{F}'_{(2b)}\}$
f_{Sgrd}	$\{\mathcal{F}'_{(1a)}\}$	$\{\mathcal{F}'_{(2b)}\}$
f_{Sstb}		

Fig. 4. (A) Composition of local agent semantics of C'_1 and C'_2. (B) Composition of local agent semantics of C''_1 with $I_{C''_1} = \{a_3\}$ and C''_2 with $I_{C''_2} = \{a_4, a_5\}$. (a) Local agent semantics of C'_1 and C'_2. (1a) - (1c) and (2a) - (2b) refer to the sub-frameworks in Figure 3. (b) Local agent semantics of C''_1 and C''_2.

Example 3. Let us consider Figure 3 for illustration. Out of all arguments, a_1 and a_2 belong to C_1, while a_6, a_7 and a_8 to C_2. Although not explicitly specified, the other three arguments are still presumed to belong to some agent(s). In this argumentation framework, we have both $T(\mathcal{F}, C_1) = \{a_4\}$ and $T(\mathcal{F}, C_2) = \{a_3, a_5\}$, which means C_1 (C_2) takes a_4 (a_3, a_5) into consideration along with its own arguments. For any other arguments, C_1 (C_2) is uncertain as to whether they may be trusted. Although neither C_1 nor C_2 is selfish or social, when they are taken together, their trusts make the coalition structure (C_1, C_2): selfish

with skeptical agent trust; and social with credulous agent trust, so that it is as though we were composing the two local agent semantics in (A) of Figure 4 if the former and those in (B) of the same figure if the latter.

5 Summary

In this paper we introduced a dynamic semantics based on sub-frameworks, and we applied it to agent argumentation. This AAA theory of ours allows for modular processing of an argumentation framework, as in [2], but the local semantics in AAA are sub-argumentation frameworks which retain the reason as to why, for instance, some argument is accepted or rejected. To obtain the local agent semantics, each agent may choose which external arguments (and attacks from them) to take into account via its agent trust function. As we explained, the agent trust can describe selfish and social agents. We showed how to obtain global label-based extensions from agents' local semantics (sub-frameworks) and the trust function of the coalition structure of the agents which derives from credulous or skeptical composition of the agents' trust functions.

As our next step, we intend three things:

1. There is a lot of research on agent argumentation and arguing about trust, and we will consider the relation between our new frameworks and existing approaches.
2. Formal analysis of the newly introduced concepts.
3. Analysis of argument coalitions in the new triple-A frameworks.

References

1. Ryuta Arisaka and Ken Satoh. Coalition Formability Semantics with Conflict-Eliminable Sets of Arguments (Extended Abstracts). In *AAMAS*, pages 1469–1471, 2017.
2. Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert W. N. van der Torre, and Serena Villata. On the input/output behavior of argumentation frameworks. *Artif. Intell.*, 217:144–197, 2014.
3. Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.
4. Beishui Liao. Toward incremental computation of argumentation semantics: A decomposition-based approach. *Ann. Math. Artif. Intell.*, 67(3-4):319–358, 2013.
5. Beishui Liao, Li Jin, and Robert C. Koons. Dynamics of argumentation systems: A division-based method. *Artif. Intell.*, 175(11):1790–1814, 2011.
6. Tjitze Rienstra, Alan Perotti, Serena Villata, Dov M. Gabbay, and Leendert W. N. van der Torre. Multi-sorted argumentation. In Sanjay Modgil, Nir Oren, and Francesca Toni, editors, *Theorie and Applications of Formal Argumentation - First International Workshop, TAFA 2011. Barcelona, Spain, July 16-17, 2011, Revised Selected Papers*, volume 7132 of *Lecture Notes in Computer Science*, pages 215–231. Springer, 2011.