Argumentation as Exogenous Coordination*

Leendert van der Torre[1] and Tjitze Rienstra[2] and Dov Gabbay[1,3,4]

[1] University of Luxembourg, Luxembourg
leon.vandertorre@uni.lu
[2] Institute for Web Science and Technologies, University of Koblenz-Landau, Germany
tjitze@gmail.com
[3] Department of Informatics, King's College London, London, United Kingdom
dov.gabbay@kcl.ac.uk
[4] Department of Computer Science, Bar Ilan University, Tel Aviv, Israel

Abstract. Formal argumentation is one of the most popular approaches in modern logic and reasoning. The theory of abstract argumentation introduced by Dung in 1995 has shifted the focus from the internal structure of arguments to relations among arguments, and temporal dynamics for abstract argumentation was proposed by Barringer, Gabbay and Woods in 2005. In this tradition, we see arguments as reasoning processes, and the interaction among them as a coordination process. We argue that abstract argumentation can adopt ideas and techniques from formal theories of coordination, and as an example we propose a model of sequential abstract argumentation loosely inspired by Reo's model of exogenous coordination. We show how the argumentation model can represent the temporal dynamics of the liar paradox and predator-prey like behaviour.

1 Introduction

The theory of abstract argumentation introduced by Dung in 1995 [14] started a new stage in the development of formal argumentation theory. In his model, the acceptance or rejection of an argument depends on the relation between the argument with other arguments, and the acceptance status of these other arguments. In contrast, traditionally argumentation was based on a formal analysis of the logical structure of arguments, and whether an argument is accepted or rejected depended only on the argument itself, not on the other arguments. In other words, in Dung's model it is no longer sufficient to point at deficiencies in an argument to reject it, but one is required to phrase the criticism itself as an argument, such that these critical arguments themselves are open for criticism as well. In abstract argumentation, we say that when an argument attacks another argument, the attacking argument itself can be attacked by a third argument. In such a case, the argument originally attacked is *defended* by the third argument, and consequently the first argument may be *reinstated*.

Formally, abstract argumentation is a graph based reasoning formalism generalising the notion of stable sets in directed graphs. As discussed in the handbook series on

* The first 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.

formal argumentation, of which the first volume appears in 2018 [8], Dung's theory constitutes a turning point for the modern stage of formal argumentation theory, much similar to the introduction of possible worlds semantics for the theory of modality. This means that nothing could remain the same as before 1995—it should be a focal point of reference for any study of argumentation, especially if the study is critical about Dung's theory. However, in modal logic, the introduction of the possible worlds semantics has led to a complete paradigm shift, both in tools and new subjects of studies, whereas this is still not fully true for what is going on in formal argumentation theory. In this paper our aim is to inspire new tools and studies for formal argumentation based on models of formal coordination, in particular the exogenous coordination language Reo [2–4].

It is not very difficult to relate abstract argumentation to the data-flow coordination language Reo, because there are various superficial similarities and differences between the two approaches. Concerning similarities, as we explain in more detail later, both are based on a graph based representation, both use graph colouring to give compositional semantics to the graphs, and consequently both can be seen as instances of causal or explanatory non-monotonic reasoning (see [10] for a modern introduction). In particular, graph colouring is used in argumentation for the key properties of admissibility and directionality, and in Reo to deal with the context sensitive behaviour of lossy channels. Moreover, argumentation graphs have been given temporal dynamics [9], and they have been extended to input/output graphs [7, 15] that reflect the flow or directionality of reasoning in logic and argumentation [20, 21]. Concerning the superficial differences, Reo has many aspects without a directly corresponding counterpart in abstract argumentation, such as stream semantics, or buffers representing memory. Likewise, abstract argumentation has aspects which do not seem to have a direct correspondence in coordination, such as complementary theories of structured argumentation.

Besides the superficial similarities and differences between the two approaches, we believe that we can define a deeper similarity between them, based on the concept of exogenous coordination. Arbab [4] defines exogenous coordination as follows.

> "Locus of coordination refers to where coordination activity takes place, classifying coordination models as endogenous or exogenous. Endogenous models, such as Linda, provide primitives that must be incorporated within a computation for its coordination. In contrast, exogenous models, such as Manifold and Reo, provide primitives that support coordination of entities from without. In applications that use exogenous models, primitives that affect the coordination of each module are outside the module itself.
>
> Endogenous models are sometimes more natural for a given application. However, they generally lead to an intermixing of coordination primitives with computation code, which entangles the semantics of computation with coordination protocols. This intermixing tends to scatter communication/ coordination primitives throughout the source code, making the cooperation model and the coordination protocol of an application nebulous and implicit: generally, there is no piece of source code identifiable as the cooperation model or the coordination protocol of an application, that can be designed, developed, debugged, maintained, and reused, in isolation from the rest of the application code. . . .

On the other hand, exogenous models encourage development of coordination modules separately and independently of the computation modules they are supposed to coordinate. Consequently, the result of the substantial effort invested in the design and development of the coordination component of an application can manifest itself as tangible "pure coordinator modules" which are easier to understand, and can also be reused in other applications." [4]

In this paper we see arguments as reasoning processes, and we characterise the interaction among such abstract argument processes as a way of coordinating arguments. In other words, we rephrase the core idea of interaction among arguments reflected by the graph based framework and language introduced by Dung in 1995 as the introduction of exogenous coordination in logic and reasoning. This reflects a separation of concerns between the logical structure of an argument and its acceptability, and facilitates the reuse of arguments as well as the reuse of argumentation frameworks.

The deeper relation between the two approaches suggests that abstract argumentation can learn from more general models of coordination, and in particular of exogenous models of coordination. Indeed, we believe it is straightforward to incorporate ideas from Reo into formal argumentation, and to find corresponding notions in informal argumentation. For example, argumentation memory is clearly present in the argumentation of people, companies, organisations, political parties, and other kinds of socially constructed entities, for example due to bounded rationality. Also in scientific argumentation only a paradigm shift leads to the rejection of conventional wisdom. In the media, daily news articles change the opinions and the arguments of the people, and reveals a dark side of spreading alternative facts and fake news. We do not claim any authority on organisational theory, philosophy of science or media sciences, but it seems clear to us that a formal study of more temporal abstract argumentation model incorporating streams of data and arguments is both natural and useful.

As an example, and a first step to bring the two approaches closer together, this paper proposes a model of sequential abstract argumentation inspired by Reo's model of exogenous coordination. This model builds on our earlier work. Multi-sorted argumentation [26] partitions the set of arguments, for example in epistemic and goal arguments, and applies different kinds of semantics to each block in the partitioning. Input/output argumentation [7] is a model of compositionality for abstract argumentation, that makes explicit how the semantics of the individual blocks can be composed into the semantics of the whole graph. In multi-agent argumentation [5], the composition of the acceptance semantics reflects a game theoretic equilibrium between the individual acceptance functions. Traditional game theoretic semantics assumes that the agents accept their arguments at the same time, just like agents in a prisonner's dilemma choose their decisions independently of each other.

Our model of temporal dynamics [9] in this paper mimics the steps of a dialogue. Like in extensive game models, we proceed step by step. The agents in a dialogue listen to the arguments the other agents accept, and decide which arguments to accept based on this information.

Given the nature of this special volume, we assume that the readers are familiar with the challenges of coordination, the concept of exogeneous coordination, the Reo coordination language, and its semantics. Moreover, we assume that not all readers are familiar with abstract argumentation, so we repeat the basic concepts and ideas.

The layout of this paper is as follows. In Section 2 we give an overview of abstract argumentation, including graph colouring and input/output argumentation, in section 3 we introduce our variant of sequential argumentation and we discuss the liar paradox, and in Section 4 we consider temporal dynamics and predator-prey like behavior [9].

2 Abstract argumentation semantics

In this section we consider abstract argumentation semantics, and in the next section we introduce sequence semantics. We first recall Dung's abstract argumentation semantics, the commonly adopted generalisation to three valued labelings, and the generalisation to input/output argumentation.

2.1 Abstract semantics

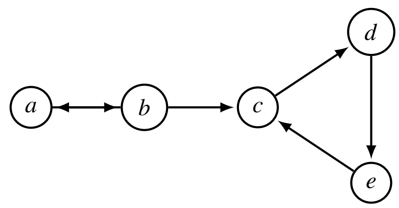
For completeness and reference below we briefly summarise Dung's abstract argumentation semantics. An argumentation framework is a directed graph whose nodes $\mathscr{A}$ are called arguments and whose edges $\mathscr{R}$ represent attack among the arguments, a kind of asymmetric inconsistency. A set $B \subseteq \mathscr{A}$ is *conflict-free* if and only if there exist no arguments a_1 and a_2 in B such that $(a_1, a_2) \in \mathscr{R}$. Argument $a \in \mathscr{A}$ is *defended* by a set $B \subseteq \mathscr{A}$ (also called a_1 is *acceptable* with respect to B) if and only if for all $a_2 \in \mathscr{A}$, if $(a_2, a_1) \in \mathscr{R}$, then there exists $a_3 \in B$ such that $(a_3, a_2) \in \mathscr{R}$. We say that a set $B \subseteq \mathscr{A}$ is *admissible*, if and only if it is *conflict-free* and defends all of its members. Based on the notion of admissible sets, Dung defines various kinds of sets of acceptable arguments called extensions. Formally, we have the following definition.

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

- *B is* conflict-free *if and only if $\nexists a_1, a_2 \in B$, s.t. $(a_1, a_2) \in \mathscr{R}$.*
- *An argument $a_1 \in \mathscr{A}$ is defended by B (equivalently a_1 is* acceptable *w.r.t. B), if and only if $\forall (a_2, a_1) \in \mathscr{R}$, $\exists \gamma \in B$, s.t. $(a_3, a_2) \in \mathscr{R}$.*
- *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 $\mathscr{A}$ 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 $\forall a_1 \in \mathscr{A} \setminus B$, $\exists a_2 \in B$ s.t. $(a_2, a_1) \in \mathscr{R}$.*

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

Example 1. Consider the argumentation framework visualized below. The complete extensions are $E_1 = \emptyset$, $E_2 = \{a\}$, and $E_3 = \{b,d\}$. The former is the unique grounded extension and the latter two are the preferred extensions, and only E_3 is a stable extension.



Dung's graph based theory has been further refined using abstract rules and assumptions, and extensions of the graph based representation have been studied as abstract dialectical frameworks. Argumentation as inference developed by Dung has been complemented by argumentation as dialogue, based on argumentation semantics as formal discussion, and argumentation schemes. In addition, computational problems have been studied, including their complexity, and implementations have been built. Formal analysis is based on a principle based approach to formal argumentation, including the use of rationality postulates to evaluate argumentation semantics. The relations between formal argumentation and other areas of formal reasoning, in particular logic, has been studied. We refer to the first volume of the Handbook of Formal Argumentation [8] for further details.

2.2 Labelling semantics

Input/output argumentation uses the labelling-based approach to the definition of argumentation semantics. 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 (Labeling). *Let* $\Lambda = \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ *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* $\mathscr{F} = (\mathscr{A}, \mathscr{R})$, *a* labelling *of* $\mathscr{F}$ *is a labelling of* $\mathscr{A}$. *The set of all* labellings *of* $\mathscr{F}$ *is denoted as* $\mathfrak{L}(\mathscr{F})$. *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 $\mathtt{in}(Lab)$ for $\{a \mid Lab(a) = \mathtt{in}\}$, $\mathtt{out}(Lab)$ for $\{a \mid Lab(a) = \mathtt{out}\}$ and $\mathtt{undec}(Lab)$ for $\{a \mid Lab(a) = \mathtt{undec}\}$.

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

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

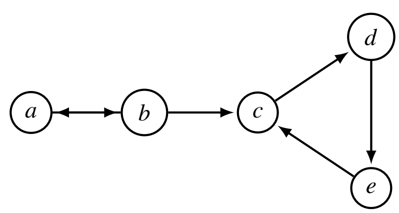
Though labelings are more general than Dung semantics, we now apply a common trick in formal argumentation: we reduce the more general notion to Dung semantics. (Just like semantics of extensions of Turing machines are reduced to Turing machines) In particular, for every Dung semantics there is a labeling based version defined by the following translation from extensions to labelings:

Definition 4 (Dung2labeling). *Given an argumentation framework $\mathscr{F} = (\mathscr{A}, \mathscr{R})$ an extension $B \subseteq \mathscr{A}$ translates to a labelling $Lab \in \mathfrak{L}_{\mathscr{A}}$ iff*

- $Lab(a) = \texttt{in}$, *if* $a \in B$,
- $Lab(a) = \texttt{out}$, *if* $a \notin B$ *and there is a* $b \in B$ *such that* $(b,a) \in \mathscr{R}$,
- $Lab(a) = \texttt{undec}$, *otherwise.*

In particular, we use $sem \in \{cmp, prf, grd, stb\}$ to denote complete, preferred, grounded, or stable labeling semantics, defined in this way in terms of the corresponding Dung semantics.

Example 2 (Continued from Ex. 1). Reconsider the argumentation framework visualized below. The complete labelings are
$L_1 = \{(a,\texttt{undec}),(b,\texttt{undec}),(c,\texttt{undec}),(d,\texttt{undec}),(e,\texttt{undec})\}$,
$L_2 = \{(a,\texttt{in}),(b,\texttt{out}),(c,\texttt{undec}),(d,\texttt{undec}),(e,\texttt{undec})\}$, and
$L_3 = \{(a,\texttt{out}),(b,\texttt{in}),(c,\texttt{out}),(d,\texttt{in}),(e,\texttt{out})\}$.
The former is the unique grounded labeling and the latter two are the preferred labelings, and only L_3 is a stable labeling.



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

Local functions define semantics for a part of the argumentation framework, called a sub-framework. In this section we repeat some basic concepts regarding local functions from Baroni *et al.*. We refer to their paper [7] for further explanations and examples. Similar notions are also defined by Liao [19, 18]. Local functions are more general than Dung semantics in the sense that they give extensions to a graph together with an input. If the input is empty, a local function coincides with a Dung semantics.

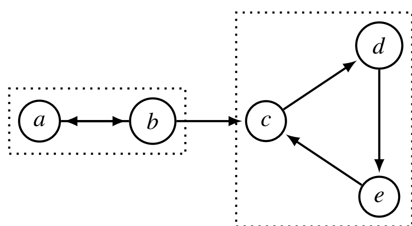
We start with the input of a subframework. Intuitively, given an argumentation framework $\mathscr{F} = (\mathscr{A}, \mathscr{R})$ and a subset B of its arguments, the elements affecting $\mathscr{F}{\downarrow_B}$, which is $(\{a \in \mathscr{A} \mid a \in B\}, \{(a_1,a_2) \in \mathscr{R} \mid a_1,a_2 \in B\})$, 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 5 (Input). *Given $\mathscr{F} = (\mathscr{A}, \mathscr{R})$ and a set $B \subseteq \mathscr{A}$, the* input *of B, denoted as B^{inp}, is the set $\{a_2 \in \mathscr{A} \setminus B \mid \exists a_1 \in B, (a_2, a_1) \in \mathscr{R}\}$, the* conditioning relation *of B, denoted as B^R, is defined as $\mathscr{R} \cap (B^{inp} \times B)$.*

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

Definition 6 (Framework with input). *An* argumentation framework with input *is a tuple $(\mathscr{F}, \mathscr{I}, L_{\mathscr{I}}, R_{\mathscr{I}})$, including an argumentation framework $\mathscr{F} = (\mathscr{A}, \mathscr{R})$, a set of arguments $\mathscr{I}$ such that $\mathscr{I} \cap \mathscr{A} = \emptyset$, a labelling $L_{\mathscr{I}} \in \mathfrak{L}_{\mathscr{I}}$ and a relation $R_{\mathscr{I}} \subseteq \mathscr{I} \times \mathscr{A}$. A* local function *assigns to any argumentation framework with input a (possibly empty) set of labellings of $\mathscr{F}$, i.e. $f(F, \mathscr{I}, L_{\mathscr{I}}, R_{\mathscr{I}}) \in 2^{\mathfrak{L}(F)}$.*

Example 3 (Continued from Ex. 2). Reconsider the argumentation framework in Example 1 and 2, together with the partitioning visualised below. The block $P_1 = \{a, b\}$ has an empty input, and block $P_2 = \{c, d, e\}$ has input$\{b\}$ with conditioning relation $\{(b, c)\}$.



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

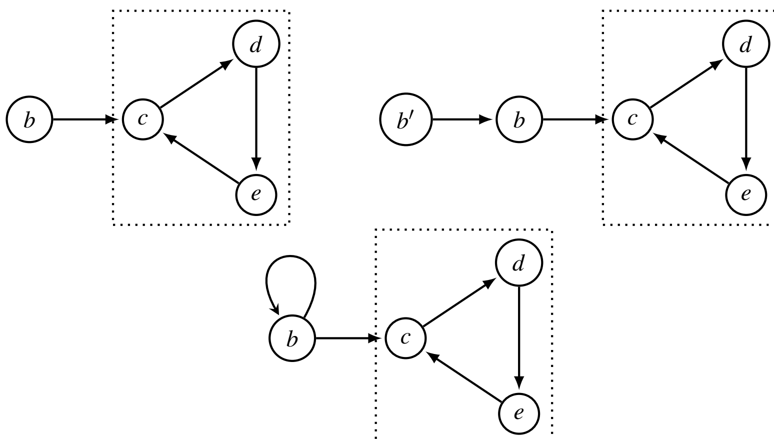
Roughly speaking, the standard argumentation framework puts $\mathscr{F}$ under the influence of $(\mathscr{I}, L_{\mathscr{I}}, R_{\mathscr{I}})$, by adding $\mathscr{I}$ to $\mathscr{A}$ and $R_{\mathscr{I}}$ to $\mathscr{R}$, and by enforcing the label $L_{\mathscr{I}}$ for the arguments of $\mathscr{I}$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

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

Definition 8 (Labeling2localfunction).
Given a semantics **S***, the* canonical local function *of* **S** *(also called local function of* **S***) is defined as* $f_{\mathbf{S}}(\mathscr{F}, \mathscr{I}, L_{\mathscr{I}}, R_{\mathscr{I}}) = \{Lab\downarrow_{\mathscr{A}} \mid Lab \in \mathbf{L_S}(\mathscr{F}')\}$*, where* $\mathscr{F} = (\mathscr{A}, \mathscr{R})$ *and* $\mathscr{F}'$ *is the standard argumentation framework w.r.t.* $(\mathscr{F}, \mathscr{I}, L_{\mathscr{I}}, R_{\mathscr{I}})$*.*

Moreover, we use *sem* $\in \{cmp, prf, grd, stb\}$ to denote complete, preferred, grounded, or stable local functions, defined in this way in terms of the corresponding labeling semantics.

Example 4 (Continued from Ex. 3). Reconsider the argumentation framework in Example 1 - 3, together with the block $P_2 = \{c, d, e\}$ with input $\{b\}$ and conditioning relation $\{(b, c)\}$. The argumentation framework with input can make argument b either `in`, `out` or `undec`, which leads to the following three standard argumentation frameworks with respect to the argumentation frameworks with input.



We refer to the paper of Baroni et al. [26, 7] for further discussion.

3 Sequential abstract argumentation

One requirement for a dynamic semantics is to be able to represent the liar paradox: If "this sentence is false" is true, then the sentence is false, but if the sentence states that it is false, and it is false, then it must be true. It is related to Epimenides paradox, Epimenides, a Cretan, said that "All Cretans are liars," and other paradoxes such as Russell's paradox. In a dynamic semantics, the truth value of the sentences toggles between true and false, and there is consequently no fixed point. In this section we show how our sequential semantics can mimic this behaviour.

Multi-agent argumentation considers a generic argumentation framework $AF = (\mathscr{A}, \mathscr{R})$ together with an arbitrary partition of $\mathscr{A}$, i.e. a set $\{P_1, \ldots, P_n\}$ such that $\forall i \in \{1, \ldots, n\}$ $P_i \subseteq \mathscr{A}$ and $P_i \neq \emptyset$, $\bigcup_{i=1\ldots n} P_i = \mathscr{A}$ and $P_i \cap P_j = \emptyset$ for $i \neq j$. Such a partition identifies the restricted argumentation frameworks $AF{\downarrow_{P_1}}, \ldots, AF{\downarrow_{P_n}}$, that affect each other with the relevant input arguments and conditioning relations as stated in Definition 5.

A multi-agent argumentation framework extends an argumentation framework with a partitioning.

Definition 9 (Multi-agent argumentation framework). *A multi-agent argumentation framework is a tuple $\mathscr{F} = (\mathscr{A}, \mathscr{R}, \mathscr{P})$ extending an argumentation framework $(\mathscr{A}, \mathscr{R})$ with a partition $\mathscr{P} = \{P_1, \ldots, P_n\}$ of $\mathscr{A}$.*

The semantics of a multi-agent argumentation framework in Example 1-4 can be based on first computing the extension of block P_1, and thereafter the extension of block P_2 using the extension of block P_1 as input. This is the basis of a well known recursive algorithm. Multi-agent argumentation raises the question what to do when the blocks attack each other? In that case, a simple recursive algorithm does not suffice. Game theory suggests two approaches:

Nash equilibrium. In case of cycles among agents, the semantics can be based on a game-theoretic equilibrium, such as for example Nash equilibria. This approach is followed by Arisaka et al. [5]. At one moment in time, the output of the agents must be identical to the input of the other agents. For example, in a prisonner's dilemma, each agent has to make a decision at the same moment without any coordination, and game theory defines states where the strategies of the agents are in a stable equilibrium.

Dialogue. Extensive games such as dialogues are based on the idea that agent act one after the other, basing their actions on the observed actions of other agents. Sequential argumentation as we consider in this paper is based on local functions, together with the idea that the output of each framework is used as input for the next step in the sequence.

A sequence semantics prescribes a set of sequences of labellings for each argumentation framework. The sequence of extensions reflects a kind of dialogue between the blocks of the partitioning.

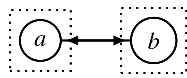
Definition 10 (Sequence semantics). *Given an argumentation framework $\mathscr{F} = (\mathscr{A}, \mathscr{R})$, a labelling-based sequence semantics* **S** *associates with $\mathscr{F}$ a set of sequences of $\mathfrak{L}(\mathscr{F})$, denoted as $\mathbf{L_S}(\mathscr{F})$.*

We use a Dung semantics to define a labeling semantics, and a labeling semantics to define an input/output semantics. Now we use an input/output semantics to define a sequence semantics. We assume that every labeling of the sequence is conflict free, though also stronger conditions may be considered. For example, one may require that every labeling of the sequence is an admissible set of F, or even a complete labeling.

Definition 11 (localfunction2sequence). *Consider a local function f. The sequence semantics of a framework F is a sequence of conflict free labelings of F, such that except for the first element of the sequence, every extension is computed using the local function f with the previous labeling of the sequence as the input.*

Again, we use $sem \in \{cmp, prf, grd, stb\}$ to denote complete, preferred, grounded, or stable sequence semantics, defined in this way in terms of the corresponding local functions.

Example 5. Consider a framework consisting of two arguments a and b attacking each other, and each argument originating from a different agent.



Since there are no cycles within a block of the partitioning, a labelling is completely determined by the input and the complete, preferred, grounded and stable labellings coincide. Three complete, preferred, grounded and stable sequences are:

$\langle\{(a,\mathtt{undec}),(b,\mathtt{undec})\},\{(a,\mathtt{undec}),(b,\mathtt{undec})\},\ldots\rangle$,
$\langle\{(a,\mathtt{in}),(b,\mathtt{out})\},\{(a,\mathtt{in}),(b,\mathtt{out})\},\ldots\rangle$,
$\langle\{(a,\mathtt{out}),(b,\mathtt{in})\},\{(a,\mathtt{out}),(b,\mathtt{in})\},\ldots\rangle$,

We can also have cyclic behaviour:

$\langle\{(a,\mathtt{undec}),(b,\mathtt{in})\},\{(a,\mathtt{out}),(b,\mathtt{undec})\},\{(a,\mathtt{undec}),(b,\mathtt{in})\},\ldots\rangle$,
$\langle\{(a,\mathtt{out}),(b,\mathtt{out})\},\{(a,\mathtt{in}),(b,\mathtt{in})\},\{(a,\mathtt{out}),(b,\mathtt{out})\},\ldots\rangle$,

If we represent the above sequences as sequences of extensions (an argument is in the extension iff it is labeled in) then we obtain the characteristic sequence of a liar paradox, which shows that

$\langle\{b\},\emptyset,\{b\},\emptyset,\ldots\rangle$

The cyclic behaviour in the the previous example occurs when the initial labelling of the sequence is itself not an admissible labelling. The following propositions show that this is no coincidence.

Proposition 1. *If a labelling in a sequence is a preferred labelling, then it is a fixed point: all following labelings in the sequence will be the same.*

Proposition 2. *If a labelling in a sequence is a complete labelling, then all following labellings will be refinements, where L_1 refines L_2, written as $L_1 \sqsubseteq L_2$, iff $in(L_1) \subseteq in(L_2)$*

4 Predator-prey models

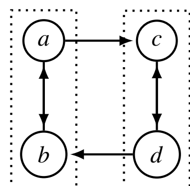
Barringer et al [9] generalise argumentation frameworks in several directions. Following various other work in formal argumentation, they allow also support relations between arguments, they allow for varying strengths of attack and support, and such strengths of attacks or support are themselves subject to attack or support. They also introduce two new ideas. First, they allow for the strengths of attack or support to be time dependent, enabling them to model the phenomenon of "Let's lie low and wait for the argument to blow away". Secondly, they examine loop-resolution in argumentation networks, and explores similarities between such loops and predator-prey models in mathematical biology.

A requirement for temporal dynamics is to mimic the predator-prey behaviour. The predator-prey equations, also known as the Lotka-Volterra equations, are a pair of first-order, nonlinear, differential equations frequently used to describe the dynamics of biological systems in which two species interact, one as a predator and the other as prey.

The populations change through time according to the pair of equations. The Lotka-Volterra system of equations is an example of a Kolmogorov model. In abstract argumentation, the requirement is to have two arguments that are accepted, then rejected, then accepted and so on.

The following example illustrates that such predator-prey behaviour can also be mimicked in our sequence semantics, without introducing numbers, support relations, attacks on attacks and so on.

Example 6. Consider a framework consisting of four arguments a, b, c and d, where the first two arguments belong to the first agent, and the latter two arguments belong to the second agent. Intuitively, the first agent can choose between accepting a or b (or none), and the second agent can choose between accepting c or d. However, these decisions are interdependent. When the first agent chooses a, the second agent no longer can choose c, and when the second agent chooses d, the first agent can no longer choose b.



Since there are loops in the argumentation frameworks of the agents, the four semantics no longer coincide. We consider the grounded semantics. In this case, given a labeling of the sequence, the following label is completely defined. If we start with a complete labeling, then all elements of the sequence are identical:

$\langle\{(a,\texttt{undec}),(b,\texttt{undec}),(c,\texttt{undec}),(d,\texttt{undec})\},\ldots\rangle$,
$\langle\{(a,\texttt{in}),(b,\texttt{out}),(c,\texttt{out}),(d,\texttt{in})\},\ldots\rangle$,
$\langle\{(a,\texttt{out}),(b,\texttt{in}),(c,\texttt{in}),(d,\texttt{out})\},\ldots\rangle$,

We can also have cyclic behaviour:

$\langle\{(a,\texttt{in}),(b,\texttt{out}),(c,\texttt{undec}),(d,\texttt{undec})\},\{(a,\texttt{undec}),(b,\texttt{undec}),(c,\texttt{out}),(d,\texttt{in})\},\ldots\rangle$,

If we represent the latter sequence as a sequence of extensions, then we obtain the characteristic sequence of a predator-prey model.

$\langle\{a\},\{d\},\{a\},\{d\},\ldots\rangle$

The same model can be used also to describe the pork cycle, hog cycle, or cattle cycle[1] in economics, describing the phenomenon of cyclical fluctuations of supply and prices in livestock markets.

5 Related work

Multi-sorted argumentation [26] considers a generic argumentation framework $AF = (\mathscr{A},\mathscr{R})$ together with an arbitrary partition of $\mathscr{A}$. Such a partition identifies the restricted argumentation frameworks $AF{\downarrow_{P_1}},\ldots,AF{\downarrow_{P_n}}$, that affect each other with the relevant input arguments and conditioning relations as stated in Definition 5.

A multi-sorted argumentation framework extends an argumentation framework with a partitioning and for each block P of the partitioning, a local function f_P.

Definition 12. *A multi-sorted argumentation framework is a tuple $\mathscr{F} = (\mathscr{A}, \mathscr{R}, \mathscr{P}, f)$ extending an argumentation framework $(\mathscr{A}, \mathscr{R})$ with a partition $\mathscr{P} = \{P_1, \ldots, P_n\}$ of $\mathscr{A}$, and a function f associating a local function f_P with every element P of $\mathscr{P}$.*

Any labelling of a restricted framework is used by f for computing the other ones: L_{P_i} plays a role in determining $L_{P_1}, \ldots, L_{P_{i-1}}, L_{P_{i+1}}, \ldots, L_{P_n}$ and vice versa. This means that $L_{P_1}, \ldots, L_{P_n}$ are "compatible" if each L_{P_i} is produced by f for $AF{\downarrow_{P_i}}$ with the input arguments P_i^{inp} labelled according to $L_{P_1}, \ldots, L_{P_{i-1}}, L_{P_{i+1}}, \ldots, L_{P_n}$. Definition 14 synthesizes all these considerations.

The extensions of a multi-sorted argumentation framework are defined as follows.

Definition 13. $\mathbf{L_S}(F) = \mathscr{U}(\mathscr{P}, AF, f)$ *where* $\mathscr{U}(\mathscr{P}, AF, f) = \{L_{P_1} \cup \ldots \cup L_{P_n} \mid L_{P_i} \in f(AF{\downarrow_{P_i}}, P_i^{inp}, (\bigcup_{j=1\ldots n, j\neq i} L_{P_j}){\downarrow_{P_i^{inp}}}, P_i^R)\}$.

Also see the recent paper of Giacomin [15], who argues that disagreements are in general heterogeneous and thus should be treated in different ways according both to their nature and to the specific agents features. Moreover, he discusses a general model of abstract argumentation based on input/output argumentation, able to handle heterogeneous disagreements by means of multiple argumentation semantics at a local level.

Baroni et al. [26, 7] aim at introducing a formal notion of semantics decomposability. To this purpose, consider a generic argumentation framework $AF = (\mathscr{A}, \mathscr{R})$ and an arbitrary partition of $\mathscr{A}$. Such a partition identifies the restricted argumentation frameworks $AF{\downarrow_{P_1}}, \ldots, AF{\downarrow_{P_n}}$, that affect each other with the relevant input arguments and conditioning relations as stated in Definition 5. Intuititively a semantics $\mathbf{S}$ is decomposable if $\mathbf{S}$ can be put in correspondence with a local function f such that:

- every labelling prescribed by $\mathbf{S}$ on AF, namely every element of $\mathbf{L_S}(F)$, corresponds to the union of n "compatible" labellings $L_{P_1}, \ldots, L_{P_n}$ of the restricted argumentation frameworks, all of them obtained applying f;
- in turn, each union of n "compatible" labellings $L_{P_1}, \ldots, L_{P_n}$ obtained applying f to the restricted frameworks gives rise to a labelling of AF.

The "compatibility" constraint mentioned above reflects the fact that any labelling of a restricted framework is used by f for computing the other ones: L_{P_i} plays a role in determining $L_{P_1}, \ldots, L_{P_{i-1}}, L_{P_{i+1}}, \ldots, L_{P_n}$ and vice versa. This means that $L_{P_1}, \ldots, L_{P_n}$ are "compatible" if each L_{P_i} is produced by f for $AF{\downarrow_{P_i}}$ with the input arguments P_i^{inp} labelled according to $L_{P_1}, \ldots, L_{P_{i-1}}, L_{P_{i+1}}, \ldots, L_{P_n}$. Definition 14 synthesizes all these considerations.

Definition 14. *A semantics* $\mathbf{S}$ *is* fully decomposable *(or simply* decomposable*) iff there is a local function f such that for every argumentation framework $AF = (\mathscr{A}, \mathscr{R})$ and every partition $\mathscr{P} = \{P_1, \ldots, P_n\}$ of $\mathscr{A}$,* $\mathbf{L_S}(F) = \mathscr{U}(\mathscr{P}, AF, f)$ *where* $\mathscr{U}(\mathscr{P}, AF, f) = \{L_{P_1} \cup \ldots \cup L_{P_n} \mid L_{P_i} \in f(AF{\downarrow_{P_i}}, P_i^{inp}, (\bigcup_{j=1\ldots n, j\neq i} L_{P_j}){\downarrow_{P_i^{inp}}}, P_i^R)\}$.

Argumentation by autonomous agents have been studied mostly in the context of strategic argumentation games, e.g. [1, 24, 17, 23, 27, 22]. An agent in negotiation dialogues as studied in [1] characterise changes in the set of accepted arguments in response to new arguments another agent introduces into his/her local scope, which, as ours, respects agents locality. In comparison, the focus of our work is more on analysing how derivation, as done by local agents, of their local semantics influences arguments acceptance globally. local agent semantics. Agents attributes are discussed in [22]. While many studies on game-theoretic argumentation games have presupposed complete information (see [16]), realistic legal examples often involve uncertainty of the belief state of other agents', and a theory that adapts to incomplete information is highly relevant.

Rahwan and Larson [25] contemplate (re)construction of an argumentation framework from the arguments in a given argumentation framework that are distributed across agents. In the construction process, the agents may or may not reveal the global outcome to be obtained varies with their decisions.

Judgement aggregation [6, 12, 11, 28] to determine acceptable arguments based on social choice theory or aggregation of argumentation frameworks [13] are being studied. While they are not the main focus of this paper, such studies become important when we deal with agents perception of other agents' local argumentation. We aim to extend our theory for that kind of a situation in a future work.

The contributions in the first volume of the Handbook of Formal Argumentation (HOFA) highlight the main innovations of this new stage of formal argumentation theory, appealing to all disciplines, including logic, computer science, law, philosophy, and linguistics. Maybe the most pressing question is how this theory of formal argumentation, developed from the area of non-monotonic logic and artificial intelligence, can be used as the foundations for informal argumentation in areas such as linguistics and law. Future volumes of the handbook series will consider extensions of Dung's theory, including numerical ones, dynamics and update, dialogue, and applications, for example in artificial intelligence, computer science, linguistics or legal reasoning. Please visit the website for more information: http://formalargumentation.org/

6 Conclusion

Dung introduced in 1995 a model of abstract argumentation focussing on the relation among arguments, in the sense that the acceptance of arguments depends on the acceptance of other agents. In his examples, arguments are derivations in logic programming, default logic, or game theory.

Many people have given a more dynamic interpretation to abstract argumentation, for example developing dialogue based decision procedures to determine whether an argument is accepted or not, or developing input/output argumentation frameworks.

Inspired by Reo, in this paper we go one step further and suggest that Dung frameworks can characterise argumentation in terms of the *interaction* among arguments, and that abstract argumentation can be characterised as the exogenous coordination of arguments. This implies that arguments themselves should not be seen as static derivations anymore, but as dynamic argumentation processes.

As an example, we showed how the argument graph can give rise not only to sets of extensions, as in Dung's semantics, but also to sequences of such extensions. Moreover, we show that the ecology interpretation of Barringer et al. can also be represented in our model without introducing numbers.

There are many issues for further study. For example, other elements of Reo can be introduced in abstract argumentation, more realistic examples can be modelled using the idea of dynamic arguments, and the formal methods of Reo can be compared to the formal techniques used in abstract argumentation.

References

1. Leila Amgoud and Srdjan Vesic. A formal analyis of the role of argumentation in negotiation dialogues. *Journal of Logic and Computation*, 5:957–978, 2012.
2. Farhad Arbab. Reo: a channel-based coordination model for component composition. *Mathematical Structures in Computer Science*, 14(3):329–366, 2004.
3. Farhad Arbab. Abstract behavior types: a foundation model for components and their composition. *Sci. Comput. Program.*, 55(1-3):3–52, 2005.
4. Farhad Arbab. Composition of interacting computations. In Dina Goldin, Scott Smolka, and Peter Wegner, editors, *Interactive Computation*, pages 277–321. Springer, 2006.
5. Ryuta Arisaka, Ken Satoh, and Leendert van der Torre. Anything you say can be used against you in a court of law. In *AICOL*, 2018.
6. Edmod Awad, Richard Booth, Fernando Tohmé, and Iyad Rahwan. Judgement Aggregation in Multi-Agent Argumentation. *Journal of Logic and Computation*, 27(1):227–259, 2017.
7. 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.
8. Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors. *Handbook of Formal Argumentation*, volume 1. College Publications, 2018.
9. Howard Barringer, Dov M. Gabbay, and John Woods. Temporal dynamics of support and attack networks: From argumentation to zoology. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Computer Science*, pages 59–98. Springer, 2005.
10. A. Bochman. *Explanatory nonmonotonic reasoning*. 2005.
11. G. Bodanza and M. Auday. Social argument justification: some mechanisms and conditions for their coincidence. In *ECSQARU*, pages 95–106, 2009.
12. Martin Caminada and G. Pigozzi. On judgement aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.
13. Sylvie Coste-Marquis, Caroline Devred, and Sébastien Konieczny. On the Merging of Dung's Argumentation Systems. *Artificial Intelligence*, 171(10-15):730–753, 2007.
14. Phan M. Dung. On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and n-Person Games. *Artificial Intelligence*, 77(2):321–357, 1995.
15. Massimiliano Giacomin. Handling heterogeneous disagreements through abstract argumentation (extended abstract). In Bo An, Ana L. C. Bazzan, João Leite, Serena Villata, and Leendert W. N. van der Torre, editors, *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference, Nice, France, October 30 - November 3, 2017, Proceedings*, volume 10621 of *Lecture Notes in Computer Science*, pages 3–11. Springer, 2017.
16. Guido Governatori, Francesco Olivieri, Antonino Rotolo, Simone Scannapieco, and Giovanni Sartor. Two Faces of Strategic Argumentation in the Law. In *JURIX*, pages 81–90, 2014.

17. Davide Grossi and W. van der Hoek. Audience-based uncertainty in abstract argument games. In *IJCAI*, pages 143–149, 2013.
18. Beishui Liao. Toward incremental computation of argumentation semantics: A decomposition-based approach. *Ann. Math. Artif. Intell.*, 67(3-4):319–358, 2013.
19. Beishui Liao, Li Jin, and Robert C. Koons. Dynamics of argumentation systems: A division-based method. *Artif. Intell.*, 175(11):1790–1814, 2011.
20. David Makinson and Leendert W. N. van der Torre. Input/output logics. *J. Philosophical Logic*, 29(4):383–408, 2000.
21. David Makinson and Leendert W. N. van der Torre. Constraints for input/output logics. *J. Philosophical Logic*, 30(2):155–185, 2001.
22. Simon Parsons and Elizabeth Sklar. How agents alter their beliefs after an argumentation-based dialogue. In *ArgMAS*, pages 297–312, 2005.
23. A. Procaccia and J. Rosenschein. Extensive-form argumentation games. In *EUMAS*, pages 312–322, 2005.
24. I. Rahwan and K. Larson. Argumentation and game theory. In *Argumentation in Artificial Intelligence*, pages 321–339. Springer, 2009.
25. Iyad Rahwan and K. Larson. Mechanism design for abstract argumentation. In *AAMAS*, pages 1031–1038, 2008.
26. Tjitze Rienstra, Alan Perotti, Serena Villata, Dov M. Gabbay, and Leendert W. N. van der Torre. Multi-sorted argumentation. In *Theorie and Applications of Formal Argumentation - First International Workshop, TAFA 2011. Barcelona, Spain, July 16-17, 2011, Revised Selected Papers*, pages 215–231, 2011.
27. R. Riveret and Henry Prakken. Heuristics in argumentation: A game theory investigation. In *COMMA*, pages 324–335, 2008.
28. Fernando Tohmé, G. Bodanza, and Guillermo R. Simari. Aggregation of attack relations: a social-choice theoretical analysis of defeasibility criteria. In *FoIKS*, pages 8–23, 2008.