A formal analysis of the AIF in terms of the ASPIC framework

Floris BEX [a], Henry PRAKKEN [b] and Chris REED [a]
[a] *Argumentation Research Group, School of Computing, University of Dundee*
[b] *Department of Information and Computing Sciences, Utrecht University*
Faculty of Law, University of Groningen

Abstract In order to support the interchange of ideas and data between different projects and applications in the area of computational argumentation, a common ontology for computational argument, the Argument Interchange Format (AIF), has been devised. One of the criticisms levelled at the AIF has been that it does not take into account formal argumentation systems and their associated argumentation-theoretic semantics, which are part of the main focus of the field of computational argumentation. This paper aims to meet those criticisms by analysing the core AIF ontology in terms of the recently developed ASPIC argumentation framework.

Keywords. ontology, argument interchange, formal argumentation framework

1. Introduction

Argumentation is a rich research area, which uses insights from such diverse disciplines as artificial intelligence, linguistics, law and philosophy. In the past few decades, AI has developed its own sub-field devoted to computational argument, in which significant theoretical and practical advances are being made. This fecundity, unfortunately, has a negative consequence: with many researchers focusing on different aspects of argumentation, it is increasingly difficult to reintegrate results into a coherent whole. To tackle this problem, the community has initiated an effort aimed at building a common ontology for computational argument, which will support interchange between research projects and applications in the area: the Argument Interchange Format (AIF) [4, 12]. The AIF's main practical goal is to facilitate the research and development of various tools for argument manipulation, argument visualization and multi-agent argumentation [4]. In addition to this, the AIF also has a clear theoretical goal, namely to provide a general core ontology that encapsulates the common subject matter of the different (computational, linguistic, philosophical) approaches to argumentation.

Although the AIF takes its inspiration from different disciplines, its roots and goals are firmly in the field of computational argument. There therefore has to be a clear connection between the AIF core ontology and computational theories of argument. However, the AIF does as of yet not fully take into account such theories; while the work that has discussed the AIF to date [13, 14] deals with issues which are important for computational argument, such as argumentation schemes [17] and dialogues [16], the examples and the general flavour of this work clearly stem from philosophical argumentation theory. Most importantly, the relation between the AIF and the various logics for argumentation and their associated argumentation-theoretic semantics (such as [5]) has not yet been clarified.

In this paper, we aim to meet the above-mentioned criticisms of the AIF by interpreting the AIF core ontology in terms of a formal (logical) argumentation theory. More specifically, we explicitly show the connection between the elements of the AIF ontology and the recently developed ASPIC framework for argumentation [9]. This framework is well-suited as a formal basis for the ontology because, like the AIF, it attempts to integrate ideas from different approaches in the literature [5, 8, 15, 10]. Furthermore, because the ASPIC framework is explicitly linked to the argumentation-theoretic semantics of [5], giving arguments expressed using the AIF ontology meaning in terms of the ASPIC framework allows the arguments to be evaluated in these semantics.

The rest of this paper is organized as follows. In section 2 we discuss the core Argument Interchange Format and give a simple example which we will refer to in the rest of the paper. Section 3 discusses the relevant parts of the ASPIC framework as set out by [9]. Section 4 formalizes the connection between the AIF and the ASPIC framework. First, we show how an AIF argumentation graph can be conceived of as an ASPIC argumentation theory (section 4.1) and then (section 4.2) we define how ASPIC arguments can be translated as an AIF argumentation graph. Section 5 concludes the paper and discusses some related and future research.

2. The Argument Interchange Format

The AIF is a communal project which aims to consolidate some of the defining work on (computational) argumentation [4]. It works under the assumption that a common vision and consensus on the concepts and technologies in the field promotes the research and development of new argumentation tools and techniques. In addition to practical aspirations, such as developing a way of interchanging data between tools for argument manipulation and visualization, the AIF project also aims to develop a commonly agreed-upon core ontology that specifies the basic concepts used to express arguments and their mutual relations. The purpose of this ontology is not to replace other (formal) languages for expressing argument but rather to serve as an interlingua that acts as the centrepiece to multiple individual reifications.

The core AIF ontology (Figure 1) falls into two natural halves: the Upper Ontology and the Forms ontology [13, 12]. In the ontology, arguments and the relations between them are conceived of as an *argument graph*. The Upper Ontology defines the language of nodes with which a graph can be built and the the Forms Ontology defines the various argumentative concepts or *forms* (e.g. argumentation schemes).

Figure 1. The Upper and Forms Ontologies of the AIF

The AIF ontology places at its core a distinction between *information*, such as propositions and sentences, and *schemes*, general patterns of reasoning such as inference or attack. Accordingly, the Upper Ontology defines two types of nodes: information nodes (I-nodes) and scheme nodes (S-nodes). Scheme nodes can be rule application nodes (RA-nodes), which denote applications of an inference rule or scheme, conflict application nodes (CA-nodes), which denote a specific conflict, and preference application nodes (PA-nodes), which denote specific preferences. Nodes are used to build an *AIF argument graph* (called argument networks by [13, 12]), which can be defined as follows:

Definition 2.1 An *AIF argument graph* G is a simple digraph (V, E) where

- $V = I \cup RA \cup CA \cup PA$ is the set of nodes in G, where I are the I-nodes, RA are the RA-nodes, CA are the CA-nodes and PA are the PA-nodes; and
- $E \subseteq V \times V \setminus I \times I$ is the set of the edges in G; and
- if $v \in N \setminus I$ then v has at least one direct predecessor and one direct successor.

We say that, given two nodes $v_1, v_2 \in V$ v_1 is a *direct predecessor* of v_2 and v_2 is a *direct successor* of v_1 if there is an edge $(v_1, v_2) \in E$.

For current purposes, we assume that a node consists of some content (i.e. the information or the name of the scheme that is being applied) and some identifier. I-nodes can only be connected to other I-nodes via S-nodes: there must be a scheme that expresses the rationale behind the relation between I-nodes. S-nodes, on the other hand, can be connected to other S-nodes directly (see Figure 2). The ontology does not type the edges in a graph; instead, semantics for edges can be inferred from the node types they connect.

In addition to the Upper Ontology, which defines the basic language for building argument graphs,[1] [13] introduced the Forms Ontology, which contains the abstract argumentative concepts. In the AIF ontology a pattern of reasoning can be an inference scheme, a conflict scheme or a preference scheme, which express a support relation (A therefore B), a conflict relation (A attacks B) and a preference relation (A is preferred to B), respectively. Scheme types can be further classified. For example, inference schemes can be deductive or defeasible and defeasible inference schemes can be subdivided into more specific argumentation schemes (e.g. Expert Opinion or Witness Testimony, see [17]). We will not explicitly define these schemes but simply assume the Forms Ontology is a set $\mathcal{F}$ which contains the relevant forms. The Forms Ontology is connected to the Upper Ontology, so that it is clear exactly what kind of form a particular node type uses (i.e. instantiates). For example, an application of an inference rule (RA node) uses an inference scheme from the Forms Ontology.

Figure 2 gives an example of an AIF argument graph, in which I-nodes are shown as rectangles and S-nodes as ellipses. The forms have been indicated above the nodes in italics. Here, the scheme for Witness Testimony (a defeasible scheme) is used to infer I_2 from I_1 and a deductive scheme is then used to subsequently infer I_3. Note that some nodes use multiple forms; I_2, for example, is the conclusion of the first inference step (that uses RA_1) but the premise of the second (that uses RA_2). RA_1 is attacked by its exception, I_4, through a Witness Bias conflict scheme. I_4 is itself attacked by I_5 and vice versa, and I_5 is preferred over I_4.

Figure 2. An AIF argument graph linked to the Forms Ontology

[1]It should be noted that, in a sense, the choice of the representational language is arbitrary. It would, for example, be perfectly acceptable to model arguments not as graphs but as sequences of sentences, as long as the information, schemes applications and the connection between them are somehow represented.

The abstract AIF ontology as presented here is purely intended as a language for expressing arguments. In order to do anything meaningful with such arguments (e.g. visualize, query, evaluate and so on), they must be expressed in a more concrete language so that they can be processed by additional tools and methods. For example, [13] reified the abstract ontology in RDF, a Semantic Web-based ontology language, which may then be used as input for a variety of Semantic Web argument annotation tools. In a similar vein, [11] have formalized the AIF in Description Logic, which allows for the automatic classification of schemes and arguments. In the current paper, one of the aims is to show how AIF argument graphs can be evaluated, that is, how a certain defeat status can be assigned to the elements of an argument graph using the argumentation-theoretic semantics of [5]. To this end, the abstract ontology needs to be reified in a general framework for formal argumentation, in this case the ASPIC framework that will be explained in the next section.

3. The ASPIC framework

The framework of [9] further develops the attempts of [1, 3] to integrate within [5]'s abstract approach the work of [8, 15, 10] on rule-based argumentation. The framework instantiates Dung's abstract approach by assuming an unspecified logical language and by defining arguments as inference trees formed by applying deductive (or 'strict') and defeasible inference rules. The notion of an argument as an inference tree naturally leads to three ways of attacking an argument: attacking an inference, attacking a conclusion and attacking a premise. To resolve such conflicts, preferences may be used, which leads to three corresponding kinds of defeat: undercutting, rebutting and undermining defeat. To characterize them, some minimal assumptions on the logical object language must be made, namely that certain well-formed formulas are a contrary or contradictory of certain other well-formed formulas. Apart from this the framework is still abstract: it applies to any set of inference rules, as long as it is divided into strict and defeasible ones, and to any logical language with a contrary relation defined over it. The framework also abstracts from whether inference rules are domain-specific (as in e.g. default logic and logic programming) or whether they express general patterns of inference, such as the deductive inferences of classical logic or defeasible argumentation schemes. In the rest of this section, the framework will be defined; an extended example is given in section 4, where we translate the graph from Figure 2 to the ASPIC framework.

The basic notion of the framework is that of an argumentation system.

Definition 3.1 [Argumentation system] An *argumentation system* is a tuple $AS = (\mathcal{L}, ^-, \mathcal{R}, \leq)$ where

- $\mathcal{L}$ is a logical language,
- $^-$ is a contrariness function from $\mathcal{L}$ to $2^{\mathcal{L}}$
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules such that $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$,
- $\leq$ is a partial preorder on $\mathcal{R}_d$.

Definition 3.2 [Logical language] Let $\mathcal{L}$, a set, be a logical language. If $\varphi \in \overline{\psi}$ then if $\psi \notin \overline{\varphi}$ then φ is called a *contrary* of ψ, otherwise φ and ψ are called *contradictory*. The latter case is denoted by $\varphi = -\psi$ (i.e., $\varphi \in \overline{\psi}$ and $\psi \in \overline{\varphi}$).

Arguments are built by applying inference rules to one or more elements of $\mathcal{L}$. Strict rules are of the form $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$, defeasible rules of the form $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$, interpreted as 'if the antecedents $\varphi_1, \ldots, \varphi_n$ hold, then *necessarily* / *presumably* the consequent φ holds', respectively. As is usual in logic, inference rules can be specified by schemes in which a rule's antecedents and consequent are metavariables ranging over $\mathcal{L}$.

Arguments are constructed from a knowledge base, which is assumed to contain three kinds of formulas.

Definition 3.3 [Knowledge bases] A *knowledge base* in an argumentation system $(\mathcal{L}, ^-, \mathcal{R}, \leq)$ is a pair $(\mathcal{K}, \leq')$ where $\mathcal{K} \subseteq \mathcal{L}$ and $\leq'$ is a partial preorder on $\mathcal{K} \setminus K_n$. Here $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_p \cup \mathcal{K}_a$ where these subsets of $\mathcal{K}$ are disjoint and

- $\mathcal{K}_n$ is a set of (necessary) *axioms*. Intuitively, arguments cannot be attacked on their axiom premises.
- $\mathcal{K}_p$ is a set of *ordinary premises*. Intuitively, arguments can be attacked on their ordinary premises, and whether this results in defeat must be determined by comparing the attacker and the attacked premise (in a way specified below).
- $\mathcal{K}_a$ is a set of *assumptions*. Intuitively, arguments can be attacked on their ordinary assumptions, where these attacks always succeed.

The following definition of arguments is taken from [15], in which for any argument A, the function Prem returns all the formulas of $\mathcal{K}$ (called *premises*) used to build A, Conc returns A's conclusion, Sub returns all of A's sub-arguments, Rules returns all inference rules in A and TopRule returns the last inference rule used in A.

Definition 3.4 [Argument] An *argument* A on the basis of a knowledge base $(\mathcal{K}, \leq')$ in an argumentation system $(\mathcal{L}, ^-, \mathcal{R}, \leq)$ is:

1. φ if $\varphi \in \mathcal{K}$ with: $\texttt{Prem}(A) = \{\varphi\}$; $\texttt{Conc}(A) = \varphi$; $\texttt{Sub}(A) = \{\varphi\}$; $\texttt{Rules}(A) = \emptyset$; $\texttt{TopRule}(A)$ = undefined.
2. $A_1, \ldots A_n \rightarrow/\Rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments such that there exists a strict/defeasible rule $\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow/\Rightarrow \psi$ in $\mathcal{R}_s/\mathcal{R}_d$.
   $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$,
   $\texttt{Conc}(A) = \psi$,
   $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$.
   $\texttt{Rules}(A) = \texttt{Rules}(A_1) \cup \ldots \cup \texttt{Rules}(A_n) \cup \{\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow/\Rightarrow \psi\}$
   $\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots \texttt{Conc}(A_n) \rightarrow/\Rightarrow \psi$

Furthermore, $\texttt{DefRules}(A) = \texttt{Rules}(A) \setminus \mathcal{R}_s$. Then A is: *strict* if $\texttt{DefRules}(A) = \emptyset$; *defeasible* if $\texttt{DefRules}(A) \neq \emptyset$; *firm* if $\texttt{Prem}(A) \subseteq \mathcal{K}_n$; *plausible* if $\texttt{Prem}(A) \not\subseteq \mathcal{K}_n$.

The framework assumes a partial preorder $\preceq$ on arguments, such that $A \preceq B$ means B is at least as 'good' as A. $A \prec B$ means that B is strictly preferred to A, where $\prec$ is the strict ordering associated with $\preceq$. The argument ordering is assumed to be 'admissible', i.e., to satisfy two further conditions: firm-and-strict arguments are strictly better than all other arguments and a strict inference cannot make an argument strictly better or worse than its weakest proper subargument. In this paper we assume that the argument ordering is somehow defined in terms of the orderings on $\mathcal{R}_d$ and $\mathcal{K}$ (definitions 3.1 and 3.3).

Because of space limitations we refer to [9] for two example definitions. The notion of an argument ordering is used in the notion of an argument theory.

Definition 3.5 [Argumentation theories] An *argumentation theory* is a triple $AT = (AS, KB, \preceq)$ where AS is an argumentation system, KB is a knowledge base in AS and $\preceq$ is an admissible ordering of the set of all arguments that can be constructed from KB in AS (below called the set of arguments on the basis of AT).

If there is no danger for confusion the argumentation system will below be left implicit.

As indicated above, when arguments are inference trees, three syntactic forms of attack are possible: attacking a premise, a conclusion, or an inference. To model attacks on inferences, it is assumed that applications of inference rules can be expressed in the object language. The general framework of [9] leaves the nature of this naming convention implicit. In this paper we assume that this can be done in terms of a subset $\mathcal{L}_R$ of $\mathcal{L}$ containing formulas of the form r or r_i. For convenience we will also use elements of $\mathcal{L}_R$ at the metalevel, as names for inference rules, letting the context disambiguate.

Definition 3.6 [Attacks]
• Argument A *undercuts* argument B (on B') iff $\texttt{Conc}(A) \in \overline{r}$ for some $B' \in \texttt{Sub}(B)$ with a defeasible top rule r.
• Argument A *rebuts* argument B on (B') iff $\texttt{Conc}(A) \in \overline{\varphi}$ for some $B' \in \texttt{Sub}(B)$ of the form $B''_1, \ldots, B''_n \Rightarrow \varphi$. In such a case A *contrary-rebuts* B iff $\texttt{Conc}(A)$ is a contrary of φ.
• Argument A *undermines* B (on φ) iff $\texttt{Conc}(A) \in \overline{\varphi}$ for some $\varphi \in \texttt{Prem}(B) \setminus \mathcal{K}_n$. In such a case A *contrary-undermines* B iff $\texttt{Conc}(A)$ is a contrary of φ or if $\varphi \in \mathcal{K}_a$.

Next these three notions of attack are combined with the argument ordering to yield three kinds of defeat. In fact, for undercutting attack no preferences will be needed to make it result in defeat, since otherwise a weaker undercutter and its stronger target might be in the same extension. The same holds for the other two ways of attack as far as they involve contraries (i.e., non-symmetric conflict relations between formulas).

Definition 3.7 [Successful rebuttal, undermining and defeat]
Argument A *successfully rebuts* argument B if A rebuts B on B' and either A contrary-rebuts B' or $A \not\prec B'$.
Argument A *successfully undermines* B if A undermines B on φ and either A contrary-undermines B or $A \not\prec \varphi$.
Argument A *defeats* argument B iff A undercuts or successfully rebuts or successfully undermines B. Argument A *strictly defeats* argument B if A defeats B and B does not defeat A.

The definition of successful undermining exploits the fact that an argument premise is also a subargument. In [9], structured argumentation theories are then linked to Dung-style abstract argumentation theories:

Definition 3.8 [*DF* corresponding to an AT] An *abstract argumentation framework* DF_{AT} *corresponding to an argumentation theory AT* is a pair $\langle \mathcal{A}, Def \rangle$ such that $\mathcal{A}$ is the set of arguments on the basis of AT as defined by Definition 3.4, and Def is the relation on $\mathcal{A}$ given by Definition 3.7.

Thus, any semantics for abstract argumentation frameworks can be applied to arguments in an ASPIC framework. In [9] it is shown that for the four original semantics of [5], ASPIC frameworks as defined above satisfy [3]'s rationality postulates (if they satisfy some further basic assumptions).

4. Analysing AIF using the ASPIC argumentation framework

In this section the connection between the core AIF ontology (section 2) and the ASPIC argumentation framework (section 3) will be clarified. This explicit connection between the informal AIF ontology and the formal ASPIC framework tells us what the AIF notation means in terms of the formal framework. While there are, of course, other ways to give meaning to the elements of the ontology, an advantage of the current approach is that by formally grounding the AIF ontology in the ASPIC framework, specific boundaries for rational argumentation are set. There are not many constraints on an argument graph, as some flexibility is needed if one wants the AIF to be able to take into account natural arguments, which are put forth by people who will not always abide by strict formal rules that govern the structure of arguments. However, one of the aims of the AIF is to provide tools for structuring arguments so that, for example, inconsistencies among arguments may be discovered. By reifying AIF argument graphs in the ASPIC framework, the arguments are expressed in a more concrete language which allows such inconsistency checking and further evaluation of complex argument graphs.

It can be asked whether the boundaries set by the ASPIC framework are the right ones. In other words, is the ASPIC framework a good argumentation logic for expressing and evaluating natural arguments? This is, of course, a big question, which cannot be given a definite answer in this paper. Nevertheless, some remarks can be made. To start with, ASPIC's tree structure of arguments fits well with many textbook accounts of argument structure and with may argument visualisation tools. Second, as argued by [9], its distinction between strict and defeasible inference rules allows a natural formalisation of argument schemes, which is an important concept from argumentation theory. Moreover, the ASPIC framework is embedded in the widely accepted semantic approach of [5] while, finally, under certain reasonable conditions it satisfies the rationality postulates of [3]. On the other hand, not all features of the AIF can be translated into ASPIC, such as reasons for contrary relations and for preferences and the boundaries to rational argumentation set by the ASPIC framework are thus limited to those forms of argumentation that can be expressed in the framework. In this respect the exercise of trying to translate the elements of the AIF ontology into the ASPIC framework also has its utility for the ASPIC framework in that it tests the limits and flexibility of this formal logical framework.

4.1. From the AIF ontology to the ASPIC framework

If we want to show the connection between the AIF ontology and the ASPIC framework, we first need to show how an AIF argument graph can be interpreted in the ASPIC theory. Since in ASPIC the argumentation framework (Definition 3.8) is inferred from an argumentation theory (Definition 3.5), all that needs to be extracted from the AIF graph is the elements of such a theory. In particular, the AIF graph does not need to directly represent the notions of an argument, argument ordering, attack and defeat. This fits the philosophy behind the AIF: graphs are as basic as possible so that they are maximally interchangeable. Properties such as defeat are *inferred properties* of an AIF graph, properties which can be inferred by some specific tool or framework that processes the graph.

Definition 4.1 Given an AIF argument graph G and a set of forms $\mathcal{F}$, an ASPIC argumentation theory AT based on G is as follows:

1. $\mathcal{L} = I \cup RA$, where $\mathcal{L}_R = RA$;
2. $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_p \cup \mathcal{K}_a$ where $\mathcal{K}_{n/p/a} = \{v \in I \mid v$ is an initial node and uses a form *axiom/premise/assumption*$\}$.
3. $\mathcal{R}_s/\mathcal{R}_d$ is the smallest set of inference rules $v_1, \ldots, v_n \rightarrow/\Rightarrow v$ (where $v_1, \ldots, v_n, v \in \mathcal{L}$) for which there is a node $v_k \in RA$ such that:
   (a) v_k uses a *deductive/defeasible scheme* $\in \mathcal{F}$; and
   (b) v_k's direct predecessors are $v_1, \ldots, v_n$ and v_k has a direct successor v.
4. $v_i \in \overline{v_j}$ iff there is a node $v_k \in CA$ such that ca has a direct predecessor v_i and a direct successor is v_j.
5. $\leq' = \{(v_i, v_j) \mid v_i, v_j \in \mathcal{K}$, there is a node $v_k \in PA$ such that pa has a direct predecessor v_i and direct successor $v_j\}$.
6. $\leq = \{(r_i, r_j) \mid r_i, r_j \in \mathcal{R}$ and $ra_i, ra_j \in RA$, there is a node $v_k \in PA$ such that v_k has a direct predecessor ra_i and direct successor $ra_j\}$.

The above definition translates elements of an AIF graph into elements of an ASPIC argumentation theory. The language of the argumentation theory consists of all I- and RA-nodes in the graph. In the case of the example from Figure 2, this means that $\mathcal{L} = \{i_1, \ldots, i_5\} \cup \{r_1, r_2\}$ (I-nodes are referred to by their identifier). $\mathcal{K}$ contains all I-nodes which are themselves not derived from other I-nodes (in the example i_1, i_4, i_5), distributed among the different subsets of $\mathcal{K}$ according to the form they use. In the example, assume that $i_1 \in \mathcal{K}_n$ (that Bob testified can not be sensibly denied) and that i_4 and i_5 are ordinary premises in $\mathcal{K}_p$. Inference rules in the ASPIC framework are constructed from the combination of RA nodes and their predecessors and successors. The type of inference rule is determined by the form that the RA node uses (the translation of schemes in the Forms Ontology to rule schemes in the ASPIC framework is left implicit). The example graph translates to the sets of inference rules as follows: $\mathcal{R}_s = \{r_2 = i_2 \rightarrow i_3\}$ and $\mathcal{R}_d = \{r_1 = i_1 \Rightarrow i_2\}$, where r_1 and r_2 correspond to ra_1 and ra_2, respectively. Contrariness is determined by whether two nodes are connected through a CA-node; in the example, $i_4 \in \overline{r_1}$, $i_4 \in \overline{i_5}$ and $i_5 \in \overline{i_4}$ (i.e. i_4 and i_5 are each other's contradictories while i_4 is a contrary of r_1). Finally, a PA-node between two initial I-nodes or between two RA-nodes translates into preferences between either elements of $\mathcal{K}$ or inference rules, respectively. In the example there is one such explicit preference: $i_4 \leq' i_5$.

Now, given the elements of the example ASPIC theory as laid out above, the following arguments can be constructed: A_1: i_1, A_2: $A_1 \Rightarrow i_2$, A_3: $A_2 \rightarrow i_3$, A_4: i_4, A_5: i_5. According to definition 3.6, A_4 undercuts both A_2 and A_3 (it attacks the application of r_1), A_4 rebuts A_5 and A_5 rebuts A_4. In order to determine defeat relations, first a preference ordering on arguments must be set. In the example, this ordering can be safely assumed to be $A_4 \prec A_5$, because $i_4 \leq' i_5$ and i_4 and i_5 are A_4 and A_5's only components. Definition 3.7 then says that A_4 defeats A_2 and A_3 (because it undercuts them), and A_5 defeats A_4 (because it successfully rebuts it). Given these defeat relations, any of [5]'s semantics can be applied. In the example, it is clear that A_1, A_2, A_3 and A_5 are acceptable: A_1 is not attacked and A_5 successfully reinstates A_2 and A_3. Argument A_4 is not acceptable because, no matter which semantics are chosen, it is not in the extension.

Most elements of an AIF argument graph can be interpreted in the ASPIC framework. However, an AIF graph may contain elements or subgraps which are not properly expressible in ASPIC. This may be due to limitations of the AIF. For example, the preferences in the graph, which are translated into orderings by clauses (4) and (5) of Definition 4.1, may not satisfy the rational constraints imposed on them by ASPIC, since users of the AIF are free to ignore these constraints. In such cases the ASPIC framework sets the rational boundaries for argumentation. However, in some cases the inability to express a part of the graph may be due to limitations of the ASPIC framework. For example, in an AIF graph PA- or CA-nodes can be supported or attacked by an I-node through an RA- or CA-node. Thus reasons for and against preferences or contrariness may be given, which is perfectly acceptable (e.g. linguistic, legal or social reasons may be given for why "married" and "bachelor" are contradictory). In its current state, the ASPIC framework does not allow such reasons to be expressed.

4.2. From the ASPIC framework to the AIF ontology

We next define a translation from ASPIC to AIF. Since the AIF is meant for expressing arguments instead of (closures of) knowledge bases, we define the translation for a given set of arguments constructed in ASPIC on the basis of a given argumentation theory. As above, the translation does not concern the notions of attack and defeat, since these can be derived from the given elements of an argumentation theory. We also assume that $\mathcal{A}$ only contains undercutters for other arguments in $\mathcal{A}$. Finally, for any function f defined on arguments we overload the symbol f to let for any set $S = \{A_1, \ldots, A_n\}$ of arguments $f(S)$ stand for $f(A_1) \cup \ldots \cup f(A_n)$.

Definition 4.2 Given a set of arguments $\mathcal{A}_{AT}$ on the basis of an ASPIC argumentation theory AT, an AIF graph G and a set of forms $\mathcal{F}$ on the basis of $\mathcal{A}_{AT}$ is as follows:

1. I is the smallest set of consisting of distinct nodes v such that:
   (a) $v \in \texttt{Conc}(\texttt{Sub}(\mathcal{A})) \setminus \mathcal{L}_R$;
   (b) if $v \in \mathcal{K}_{n/p/a}$ then v uses a form *axiom/premise/assumption* $\in \mathcal{F}$.
2. RA is the smallest set consisting of distinct nodes v for each rule r in $\texttt{Rules}(\mathcal{A})$, where if $r \in \mathcal{R}_{s/d}$ then v uses a *deductive scheme/defeasible scheme* $\in \mathcal{F}$, respectively (we say that v corresponds to r).
3. CA is the smallest set consisting of distinct nodes v for each pair $\varphi, \psi \in \texttt{Conc}(\texttt{Sub}(\mathcal{A}))$ and $\varphi \in \overline{\psi}$ (we say that v corresponds to (φ, ψ));
4. PA is the smallest set consisting of distinct nodes v for each a pair (k, k') in $\leq'$ such that $k, k' \in \texttt{Prem}(\mathcal{A})$ and for each pair (r, r') in $\leq$ such that $r, r' \in \texttt{Rules}(\mathcal{A})\}$ (we say that v corresponds to (k, k') or to (r, r'));
5. E is the smallest set such that for all v, v' in G:
   (a) If $v \in I$ and $v' \in RA$ and v' corresponds to r, then:
      i. $(v, v') \in E$ if v is an antecedent of r;

ii. $(v', v) \in E$ if v is the consequent of r;

(b) If $v, v' \in RA$, v corresponds to r and v' corresponds to r', then $(v, v') \in E$ if r' (as a wff of $\mathcal{L}_R$) is the consequent of r;

(c) If $v \in I \cup RA$ and $v' \in CA \cup PA$ and v' corresponds to (φ, ψ), then:

i. $(v, v') \in E$ if $v = \varphi$;
ii. $(v', v) \in E$ if $v = \psi$.

The above definition builds an AIF graph based on the elements of an ASPIC argumentation theory. The I-nodes consist of all the premises and conclusions of an argument in $\mathcal{A}$ (denoted by $\texttt{Conc}(\texttt{Sub}(\mathcal{A}))$). In the example (see AT as defined below definition 4.1 and Figure 2), there are five I-nodes based on the formulas $\{i_1, \ldots, i_5\}$. The set of RA-nodes consist of all inference rules applied in an argument in $\mathcal{A}$; the type of inference rule determines which form an RA-node uses. In the example, there are two inference rules, r_1 and r_2, which corresponding to the two RA-nodes ra_1 and ra_2 that use a defeasible and a deductive scheme, respectively. CA nodes correspond to conflicts between formulas occurring in arguments in $\mathcal{A}$ as determined by the contrariness relation. In Figure 2, the nodes ca_1, ca_2, ca_3 are based on the contrariness between i_4 and i_5 and i_4 and r_1. PA-nodes correspond to the preferences in AT between the rules used in arguments in $\mathcal{A}$ (i.e. a subset of $\leq$) or between the premises of arguments in $\mathcal{A}$ (a subset of $\leq'$). In the example AT there is only one such preference, namely $i_4 \leq' i_5$, which translates into pa_1 in Figure 2. Since the argument ordering $\preceq$ of AT is defined in terms of $\leq$ and $\leq'$, it is not part of the AIF graph.

The edges between the nodes are determined in terms of the relations between the corresponding elements in the AT. I-nodes representing an inference rule's antecedents and consequents are connected to the RA-node corresponding to the rule (viz., for example, the edges from i_1 to ra_1 to i_2 in Figure 2). Reasons for inference rules can be appropriately translated as links from RA-nodes to RA-nodes: condition 5b says that for any rule r in an argument with as its conclusion another rule $r' \in \mathcal{L}_R$, the RA-node corresponding to r is connected to the RA-node corresponding to r'. In this way, an argument that concludes that an inference rule should be applied (e.g. a reason for why there is no exception) can be expressed. Links from or to PA- and CA-nodes are connected to I- and RA-nodes according to the preference and contrariness relations in AT. For example, the edges from i_5 to pa_1 to i_4 are based on the fact that $i_4 \leq' i_5$. An undercutter is expressed as a link from the conclusion of the undercutter (an I-node, i_4 in the example) to a CA-node (ca_1) and a link from this CA-node to the RA-node denoting the undercut rule (ra_1). Definition 4.2 does not define the translation of edges between, for example, CA-nodes to CA-nodes, which are needed to express reasons against contrariness relations, as the ASPIC framework cannot express such reasons.

5. Conclusions and future research

In this paper we have shown how argument graphs as defined by the AIF can be formally grounded in the ASPIC argumentation framework. We have given the AIF ontology a sound formal basis and demonstrated how a formal framework can aid in tracing possible inconsistencies in a graph. Because of the formal scope of the ASPIC framework, we have also implicitly shown the connection between the AIF and other formal argumentation frameworks. In addition to the ASPIC framework's obvious relation to [5, 8, 15, 10], several other well-known argumentation systems (e.g. [2]) are shown by [9] to be special cases of the ASPIC framework. The connection between the AIF and ASPIC can therefore be extended to these systems. A topic for future research is to see what the relation is between the AIF and other formal frameworks that fall outside the scope of the ASPIC framework; this would also further clarify the relation between the ASPIC framework and these other frameworks. Thus, one of the main theoretical aims of the AIF project, namely integrate various results into a coherent whole, is realized.

The paper shows that a relatively simple AIF argument graph contains enough information for a complex formal framework such as ASPIC to work with. Information that is not contained in the graph, such as defeat relations, can be inferred from the graph as desired. This conforms to the central aim of the AIF project: the AIF is intended as a language for expressing arguments rather than a language for, for example, evaluating or visualizing arguments. That said, the discussion on what should be explicitly represented in the graph and what should count as an inferred property is by no means settled. In this regard, it would be interesting to explore how and if the AIF can be directly connected to abstract argumentation frameworks, which have the notion of argument as one of its basic components. One possibility is to introduce new nodes – A-nodes perhaps – which link to all the components (I-nodes, RA-nodes, etc.) from which the argument is composed. An implementation of this idea has been trialled in a tool for computing acceptability semantics.[2] One problem, however, is how to characterize A-nodes precisely – they seem to have some of the character of an I-node, but on the other hand, could be interpreted just as sets of properties of other nodes. Given both these ontological problems and further challenges in implementation we currently leave A-nodes to future work.

Some properties of argumentation represented in an AIF graph cannot be expressed in the ASPIC framework, in particular reasons for contrariness relations and preferences. Some of these shortcomings are being addressed: [7] present an extension of the ASPIC system along the lines of [6], in which attacks on attacks can be modelled with arguments about preference relations between premises or defeasible inference rules. It seems that the preference statements in such arguments can be extracted from an AIF graph as follows. First for each PA-node $v_k \in PA$ with direct predecessor $v_i \in I \cup RA$ and direct successor $v_j \in I \cup RA$, a first-order formula $v_i \leq v_j$ is added to $\mathcal{L}$. Then each such node v_k that is supported by I-nodes through an RA-node v_l, the translation of v_l into an element of $\mathcal{R}$ has conclusion $v_i \leq v_j$. Finally, the appropriate naming conventions should be defined in $\mathcal{L}$ to make sure that the first-order terms v_i and v_j denote the right formulas in $\mathcal{L}$ or rules in $\mathcal{R}$. In our future work we intend to fully develop these ideas so as to keep the translation functions between the AIF and ASPIC up-to-date with new versions of the ASPIC framework.

Finally, a necessary topic for future research and development is to further test the limits of the current ASPIC reification of the AIF ontology by considering less trivial examples of natural argument. **This addresses reviewer 1's scalability concerns and may provide a nice bridge to Chris' concluding thoughts on connecting arg mapping tools to formal frameworks: to test our translation functions, we can use the large corpus of arguments constructed in Araucaria, which are now also in AIF (AraucariaDB -> ArgDB -> AIFDB -> "ASPICDB")**

[2] The tool is called OVA-gen and is accessible online at *http://ova.computing.dundee.ac.uk/ova-gen/*

References

[1] L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney, S. Parsons, H. Prakken, J. van Veenen, and G.A.W. Vreeswijk. Final review and report on formal argumentation system. Deliverable D2.6, ASPIC IST-FP6-002307, 2006.

[2] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[3] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171:286–310, 2007.

[4] C.I. Chesñevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. Towards an argument interchange format. *The Knowledge Engineering Review*, 21:293–316, 2006.

[5] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and *n*–person games. *Artificial Intelligence*, 77:321–357, 1995.

[6] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173:901–934, 2009.

[7] S. Modgil and H. Prakken. Reasoning about preferences in structured argumentation frameworks. These proceedings.

[8] J.L. Pollock. Justification and defeat. *Artificial Intelligence*, 67:377–408, 1994.

[9] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1, 2010. to appear.

[10] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 7:25–75, 1997.

[11] I. Rahwan, I Banihashemi, C. Reed, Walton D., and S. Abdallah. Representing and classifying arguments on the semantic web. *Knowledge Engineering Review*, 2010. Accepted for publication.

[12] I. Rahwan and C. Reed. The argument interchange format. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*. Springer, 2009.

[13] I. Rahwan, F. Zablith, and C. Reed. Laying the foundations for a world wide argument web. *Artificial Intelligence*, 171:897–921, 2007.

[14] C. Reed, S. Wells, J. Devereux, and G. Rowe. Aif+: Dialogue in the argument interchange format. In Ph. Besnard, S. Doutre, and A. Hunter, editors, *Computational Models of Argument: Proceedings of COMMA-2008*, pages 311–323. IOS Press, 2008.

[15] G.A.W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90:225–279, 1997.

[16] D.N. Walton. *Logical dialogue-games and fallacies*. University Press of America, Inc., Lanham, MD., 1984.

[17] D.N. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, Cambridge, 2008.