On Logical Specifications of the Argument Interchange Format

Floris Bex[a,*] Sanjay Modgil[b] Henry Prakken[c] Chris Reed[a]

[a]*School of Computing, University of Dundee*
[b]*Department of Informatics, King's College London*
[c]*Department of Information and Computing Sciences, Utrecht University & Faculty of Law, University of Groningen*

Abstract

The Argument Interchange Format (AIF) has been devised in order to support the interchange of ideas and data between different projects and applications in the area of computational argumentation. In order to support such interchange, an abstract ontology for argumentation is presented, which serves as an interlingua between various more concrete argumentation languages. In this paper, we aim to give what is essentially a logical specification of the AIF ontology by mapping the ontology onto the logical ASPIC^{+} framework for argumentation. We thus lay foundations for interrelating formal logic-based approaches to argumentation captured by the ASPIC^{+} framework and the wider class of argumentation languages, including those that are more informal and user-orientated.

Keywords: argument ontology, argumentation logic

*corresponding author: School of Computing, University of Dundee, Dundee DD1 4HN
florisbex@computing.dundee.ac.uk

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 [4], 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 core ontology for computational argument, which will support interchange between research projects and applications in the area: the Argument Interchange Format (AIF) [13, 31]. 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 [13]. In addition to this, the AIF also has a theoretical goal, namely to provide an abstract ontology that encapsulates the common subject matter of the different (computational, linguistic, philosophical) approaches to argumentation.

Computational models of argument span a broad range of techniques including mathematically grounded models of computing acceptability of abstract arguments and logically grounded models of computing entailment over a set of structured arguments (see [29]); computational methods of information extraction and corpus processing specialised to argumentative texts [25]; computational systems of discourse processing [39], natural language understanding, and natural language generation [33] developed for monologic and dialogic argument; and many others. AIF strives to support interaction with all of these models.

A weakness of the AIF ontology with regards to computational models of argument is that the relation between the ontology and the various logics for argumentation [29] and their associated argumentation-theoretic semantics (e.g. [14]) has, until now, never been fully clarified (except in [8], of which the present paper is an extension). In this paper, we aim to make this relation explicit by showing how an abstract, generic specification of the AIF ontology can be mapped to a a logical framework for argumentation. More specifically, we propose a set of translation functions between the AIF core ontology and the ASPIC$^+$ framework [27]. Thus, the AIF ontology becomes formally grounded in a logical framework that instantiates the argumentation-theoretic semantics of [14] and at the same time the ASPIC$^+$ framework is placed in the wider spectrum of not just logical but also philosophical and linguistic approaches to argumentation.

The ASPIC$^+$ framework integrates ideas from several approaches in the literature and adopts an intermediate level of abstraction between the abstract approach to argumentation inspired by [14] and more concrete logics such as those developed in [6, 15, 28]. The framework has recently been extended, yielding the E-ASPIC$^+$ framework [22], which not only instantiates Dung's abstract framework [14] but also its recent extension to accommodate argumentation about preferences [20]. Our choice of this framework is motivated by the fact that it has been shown [27] to subsume, or at least closely approximate, other important work on logics for argumentation such as [11, 42]. More recently, [23] have further shown the connection to classical logic approaches to argumentation [6] (including those that accommodate preferences [3, 2]) and [41] have shown how arguments from Carneades [16], which incorporates an explicit notion of burden of proof, can be interpreted in ASPIC$^+$. A translation of the AIF ontology to the language of ASPIC$^+$ therefore provides us with information that can be easily used by other systems for computational argumentation. Furthermore, the translation functions themselves can in a sense be viewed as generic, in that they can be used to translate the AIF arguments to any logical framework that uses similar terminology and concepts to the ASPIC$^+$ framework.

The AIF ontology is only meant to encapsulate concepts that are needed to *express* arguments, not concepts that are needed to *compute* properties of these arguments or otherwise process these arguments. Such processing is possible if we express or implement the AIF ontology in more concrete languages; for example, [30] formalised the AIF ontology in Description

Logic, which allows for the automatic classification of schemes and arguments. In a similar vein, expressing AIF arguments in ASPIC$^+$ further adds to the possibilities of processing and engaging with AIF argument resources. The Argument Web [32], currently implemented as the AIFdb [18][1], is an argumentation corpus based on a Database Schema specification of the AIF that at the time of writing contains a growing set of around 2,000 non-dialogical and dialogical arguments built using tools such as Araucaria [34] and Rationale [5], OVA[2] [38] and Arvina [19]. By mapping the concepts in the the AIF ontology to the concepts in the language of ASPIC$^+$, we can evaluate the acceptability of these arguments, and this information about arguments' acceptability can then be fed back to other AIF-based tools, such as the visualiser for abstract argumentation frameworks OvaGen [38]. The possibility is thus created for computational models of argumentation to engage with large corpora of natural argument.

The rest of this paper is organized as follows. In Section 2 we discuss the core AIF ontology. We give a specification and discuss some issues regarding conflict and preference, which were only marginally touched upon in previous work. Section 3 discusses the relevant parts of the ASPIC$^+$ framework as set out by [27], and the more recent extension E-ASPIC$^+$ in [22]. In Section 4 the connection between the AIF ontology and the basic ASPIC$^+$ framework is formalized. Sections 4.1 and 4.2 respectively formalise the translation from AIF to ASPIC$^+$ and vice versa, and section 4.3 presents formal results with respect to the information-preserving properties of the translation functions. Section 5 then similarly shows two way translations between the AIF ontology and the extended E-ASPIC$^+$ framework of [22], and shows the information-preserving properties of the translation functions. In Section 6, we briefly discuss issues that arise when translating AIF representations of arguments constructed in an argument diagramming tool (*Rationale* [5]) to the ASPIC$^+$ framework, in order to evaluate the acceptability of the diagrammed arguments. Section 7 concludes the paper and discusses related and future research. Finally, the appendix contains proofs for the formal results with respect to the identity-preserving properties of the translation functions.

2 The Argument Interchange Format

The AIF is a communal project which aims to consolidate some of the defining work on computational argumentation [13]. Its aim is to facilitate a common vision and consensus on the concepts and technologies in the field so as to promote 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, a common core ontology for expressing argumentative information and relations is also developed. Thus, the AIF ontology aims to provide a bridge between linguistic, logical and formal (i.e. concerned with form) models of argument and reasoning.

On the one hand, the AIF ontology has a close relation to mark-up languages such as AML (the derivative of UML used by Araucaria [34]) and the Text Encoding Initiative specification (see `http://www.tei-c.org/`), which gives it the flexibility to handle arguments expressed naturally in text. On the other hand, it is possible to define relations to classical and non classical logics, as section 4 below demonstrates, allowing various types of automated processing to be run over the logical (and thence abstract) structures underlying arguments. And finally, it allows expression of argument structures such as those developed in philosophy and pragmatics focusing on schematic patterns [10], dialogical connections [21, 35, 43], and so on. AIF is unique in providing explicit support and theoretical connection to these three approaches to argumentation.

A common core ontology for argumentation is interesting for a number of reasons. On the practical side, the AIF ontology as an interlingua drastically reduces the number of translation

[1] `http://www.arg.dundee.ac.uk/AIFdb/`.
[2] `http://www.arg.dundee.ac.uk/ova`

functions that are needed for the different argumentation formats (i.e. implementations of individual argumentation languages) to engage with each other; only translation functions to some implementation of the core AIF ontology have to be defined (i.e., n instead of n^2 functions for n argumentation formats). On a more theoretical level, the abstract AIF ontology acts as a conceptual anchoring point for the various languages, which improves the exchange of ideas between them. By providing a strict graph-theoretic representation, the AIF ontology provides a frame of reference accessible to theories of argument founded upon first order logics, higher order and non-classical logics, and theories of argument developed in epistemological contexts, linguistic contexts, and theories applied in pedagogy, law, and so on.

A common frame of reference, however, does not obviate the problem of commonness of meaning, for it still presupposes that the developers of the various argumentation theories have some sort of common understanding of the AIF core ontology. In order to promote this common understanding, the core ontology should be kept as basic as possible and the various elements of the ontology should be clearly defined. [31] note that due to the nature of the AIF project it is unavoidable that the ontology – and thus its common interpretation – will change over time. However, by having more translations and thus more references available the common understanding of the AIF ontology will be further improved. Furthermore, the AIF project does not aim to tie applications or research projects to a particular format or interlingua. In some cases, for example when comparing two logical systems, it might be more sensible to provide a direct translation between the two systems which focuses on computational properties of those systems (as is the case in, e.g., [41]). In a sense, the AIF ontology can be understood as a *tool* for development of interchanges because it can, as it were, provide a meeting point or forum for various researchers and application developers to define translation functions that facilitate interchange.

2.1 The AIF Core Ontology

The AIF core ontology is first and foremost an abstract, high-level specification of argumentative information and the relations of inference, conflict and preference between this information.[3] The core ontology is intended as a conceptual model of arguments and the schemes or patterns arguments generally follow. It defines arguments and their mutual relations as typed graphs [13, 32], which is an intuitive way of representing argument in a structured and systematic way without the formal constraints of a logic [13]. Furthermore, this high-level description of the AIF ontology thus also meets [17]'s criterion of 'minimal encoding bias' for ontologies, which states that a conceptualisation should be specified at the knowledge level without depending on a particular symbol-level encoding.

The high-level description of the AIF ontology can be made more concrete in individual specifications that make use of particular formalisms. Examples of such specifications of the AIF are the OWL-Description Logic specification proposed by [30], the RDFs specification as discussed by [32] and the Database Schema specification outlined in [7]. These separate specifications of the AIF ontology emphasise and de-emphasise and include and exclude different aspects of the high-level description according to the nature of the language used. So, for example, an AIF specification in Description Logic focusses on classifying types of arguments according to the argumentation scheme used, whilst the Database Schema specification does not include such an explicit hierarchical structure of argument types. A translation or mapping of the AIF ontology onto the language used by the ASPIC$^+$ framework for representing arguments concretisizes the AIF specification in a formal logical language that can then be used to compute, among other things, (defeasible) logical consequence and the acceptability of arguments.

[3] The name Argument Interchange *Format* is in this respect somewhat misleading, as it seems to imply that AIF is a file format, whereas the AIF ontology can be implemented in a number of specific formats (XML, DOT, SQL). However, the name is retained for historical reasons.

2.2 Description of the AIF Core Ontology

The AIF core ontology falls into two natural halves [32, 31]: the Upper Ontology and the Forms Ontology. The Upper Ontology defines the basic building blocks of AIF argument graphs, nodes and edges. The Forms Ontology allows us to type the elements of AIF graphs in terms of argumentation-theoretical concepts such as inference, conflict and so on. Nodes can be used to build argument-graphs (see Definition 2.1) and these nodes then *fulfil* (i.e. instantiate) specific forms, such as inference schemes, from the Forms Ontology. Note that in this paper, we do not commit to any particular formalisation of how fulfilment works ([32], for example, define so-called form-nodes, or F-nodes, and denote fulfilment through edges connecting S-nodes and F-nodes and [30] define schemes as classes in OWL Description Logic and let the machinery of DL handle fulfilment).

In Figure 1, an overview is given of the main description of the AIF ontology. White nodes define the classes (concepts) in the Upper Ontology whilst grey nodes define those in the Forms Ontology. The arrows denote the different relations between the classes in the ontology: dotted arrows are *fulfils* relations and normal arrows are *is a* (class inclusion) relations. So, for example, the class of *inference schemes* is a subclass of the class of *schemes* and RA-nodes (Rule Applications) fulfil *inference schemes*.

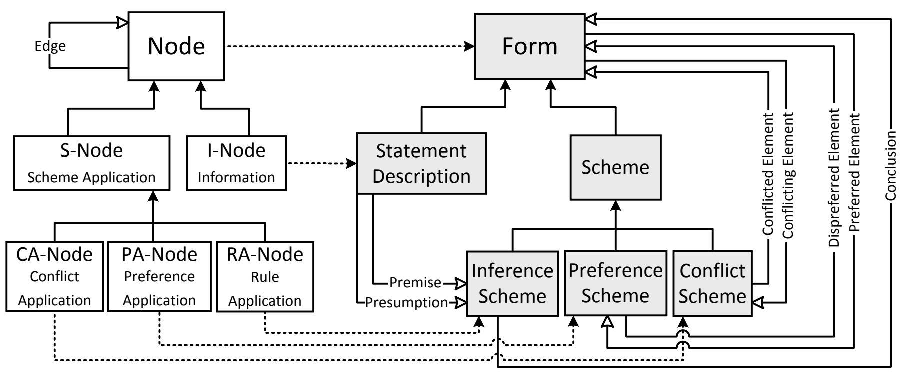

Figure 1: The AIF core ontology description. Dotted arrows are *fulfils* relations and normal arrows are *is a* relations (unless indicated otherwise).

The Upper Ontology places at its core a distinction between *information*, such as propositions and sentences, and applications of *schemes*, general patterns of reasoning such as inference or conflict. Accordingly, the Upper Ontology describes two types of nodes for building argument graphs: information nodes (I-nodes) and scheme application nodes (S-nodes), as well as an *edge* relation for the edges between nodes. S-nodes can be rule application nodes (RA-nodes), which denote specific inference relations, conflict application nodes (CA-nodes), which denote specific conflict relations, and preference application nodes (PA-nodes), which denote specific preference relations. As [13] notes, nodes can have various attributes (e.g. `creator`, `date`). In the current AIF specification, a node consists of an identifier and some content (i.e. the information or the specific scheme that is being applied).

The Forms Ontology defines the schemes and types of statements commonly used in argumentation. The cornerstones of the Forms Ontology are schemes: inference, conflict and preference are treated as genera of a more abstract class of schematic relationships [10], which allows the three types of relationship to be treated in more or less the same way, which in turn greatly simplifies the ontological machinery required for handling them. Thus, inference schemes, conflict schemes and preference schemes in the Forms Ontology embody the general principles expressing how it is that q is inferable from p, p is in conflict with q, and p is preferable to q, respectively. The individual RA-, CA- and PA-nodes that fulfil these schemes then capture the passage or the process of actually inferring q from p, conflicting p with q and preferring p to q, respectively.

The nodes from the Upper Ontology can be used to build an *AIF argument graph* (called argument networks by [32, 31]), as follows.

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

1. $V = I \cup RA \cup CA \cup PA$ 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
2. $E \subseteq V \times V \setminus I \times I$ is the set of the edges in G. Any edge $e \in E$ is assumed to have exactly one type from among the following: *premise*, *conclusion*, *preferred element*, *dispreferred element*, *conflicting element*, *conflicted element*; and
3. if $v \in V \setminus I$ then v has at least one direct predecessor and one direct successor; and
4. if $v \in RA$ then v has at least one direct predecessor via a *premise* edge and exactly one direct successor via a *conclusion* edge; and
5. if $v \in PA$ then v has exactly one direct predecessor via a *preferred element* edge and exactly one direct successor via a *dispreferred element* edge; and
6. if $v \in CA$ then v has exactly one direct predecessor via a *conflicting element* edge and exactly one direct successor via a *conflicted element* edge.

We say that, given two nodes $v_1, v_2 \in V$, v_1 is a *predecessor* of v_2 and v_2 is a *successor* of v_1 if there is a path in G from v_1 to v_2, and v_1 is a *direct predecessor* of v_2 and v_2 is a *direct successor* of v_1 if there is an edge $(v_1, v_2) \in E$. A node v is called an *initial node* if it has no predecessor.

Condition 2 states that I-nodes can only be connected to other I-nodes via S-nodes, that is, 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, e.g., Figures 3, 4). Condition 3 ensures that S-nodes always have at least one predecessor and successor, so that (a chain of) scheme applications always start and end with information in the form of an I-node.

Notice that in Definition 2.1 the edges are typed to indicate what the role of one node is with respect to another node. These edge types are defined in the Forms Ontology for each of the schemes and forms they connect (see the named arrows with the open heads in Figure 1). For example, a node that fulfils an inference scheme can have a predecessor via a *premise* or *presumption* edge and a node that fulfils a conflict scheme has a *conflicting element* predecessor and a *conflicted element* successor. Conditions 4 – 6 in Definition 2.1 state the specific constraints on edges in an argument graph: inference applications (RA-nodes) always have at least one *premise* and at exactly one *conclusion* (4), preference applications are always between two distinct nodes representing the preference's *preferred element* and *dispreferred element* (5) and conflict applications always have exactly one *conflicting element* and one *conflicted element* (6).

Inference schemes in the AIF ontology are similar to the rules of inference in a logic, in that they express the general principles that form the basis for actual inference. They can be deductive (e.g. the inference rules of propositional logic) or defeasible (e.g. [45]'s argumentation schemes). One example of an inference scheme is that of Defeasible Modus Ponens [27, 28], of which the *premises* are the minor premise φ and the major premise $\varphi \rightsquigarrow \psi$ (here, $\rightsquigarrow$ is a connective standing for defeasible implication) and the conclusion is ψ, where φ and ψ are meta variables ranging over well formed formulae in some language. Figure 2 shows an actual argument based on this scheme, represented in the AIF ontology. The scheme is indicated next to the RA-node *ra2* representing the application of the scheme and the edges show their respective types.

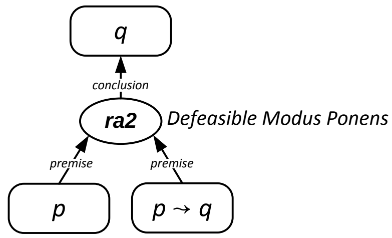

Figure 2: An AIF argument-graph

In Figure 2, the fact that q is inferable from p is represented in the object layer as the (defeasible) conditional $p \rightsquigarrow q$. In line with a long tradition in argumentation theory and non-monotonic logic (e.g. [45, 15, 36]), such specific knowledge can be modelled as inference rules itself, that is, as an inference scheme in the Forms Ontology. Take, for example, the inference scheme for Argument from Expert Opinion [45]:

Scheme for Argument from Expert Opinion
premises: E is an expert in domain D, E asserts that P is true, P is within D;
conclusion: P is true;
presumptions: E is a credible expert, P is based on evidence;
exceptions: E is not reliable, P is not consistent with the testimony of other experts.

An argument based on this scheme is rendered in Figure 3. Thus, specific (but still generalizable) knowledge can be modelled in the AIF in a principled way using argumentation schemes, for which we can assume, for example, a raft of implicit assumptions which may be taken to hold and exceptions which may be taken not to hold. Note that the AIF ontology itself does not legislate which schemes are in the Forms Ontology and the exact structure of these schemes; rather, this depends on the inference rule schemes or argumentation schemes that a particular specification of the AIF ontology uses.

Like inference, conflict is also generalizable. General conflict relations, which may be based on logic but also on linguistic or legal conventions, can be expressed as *conflict schemes* in the Forms Ontology. All conflict schemes have two elements: one element that "conflicts" and another one that "is conflicted"; symmetry is not automatically assumed so that for a symmetrical conflict a scheme has to be applied twice.[4]

As an example of a conflict scheme, take the scheme for Conflict From Expert Unreliability, which states that that the fact that an expert is unreliable is in conflict with the inference based on the Expert Opinion scheme. In other words, the *conflicting element* of this scheme is 'E is not reliable' and the *conflicted element* is the Scheme for Argument from Expert Opinion. Figure 3 shows the application of the conflict scheme Expert Unreliability, which here attacks the application of the Expert Opinion inference scheme as represented by the node *ra12* (i.e. the fact that the expert e_1 is not reliable is in conflict with the fact that p is inferred from the premises).

While inference and conflict allow us to build arguments and provide counterarguments, in many contexts a choice needs to be made as to which of the arguments is better or stronger. This can be expressed using preferences. In the AIF ontology, preference follows the now-familiar pattern that inference and conflict also follow: we assume a set of preference schemes in the Forms Ontology, which express principles for why certain (types of) information or schemes are preferable to others. A preference scheme contains a *preferred element*, the information or scheme that is preferred, and a *dispreferred element*, the information or scheme that the former

[4] Roughly, the *conflicting element* can be seen as the proposition that attacks and the *conflicted element* as the proposition that is attacked. However, because the notion of "attack" already has its own meaning in theories of computational argumentation, we are here forced to use the (rather cumbersome) terms *conflicting element* and *conflicted element*.

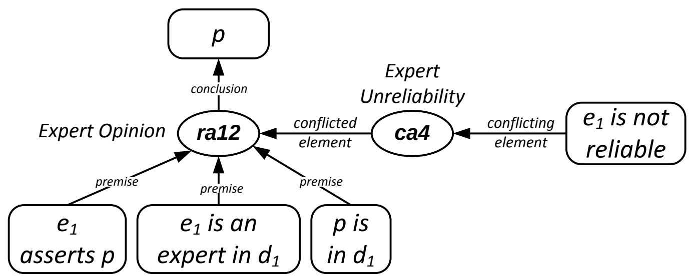

Figure 3: Conflict from Unreliability

is preferred to.

Preference schemes can define preferences between other schemes, for instance, inference schemes. As an example, consider an inference scheme for general knowledge [9], with as its premise 'It is general knowledge that P' and its conclusion 'P'. Now, we might want to say that, in general, arguments based on expert opinion are preferable to those based on general knowledge. This can be represented as a preference scheme with as its *preferred element* the inference scheme for Expert Opinion and as its *dispreferred element* the inference scheme for General Knowledge. The preference scheme can then be applied as in Figure 4. Notice that in this case the actual inference based on the Expert Opinion scheme (*ra12*) is preferred over the inference based on the General Knowledge scheme (*ra13*), because the preference scheme expresses that generally, inferences based on Expert Opinion are preferable over inferences based on General Knowledge.

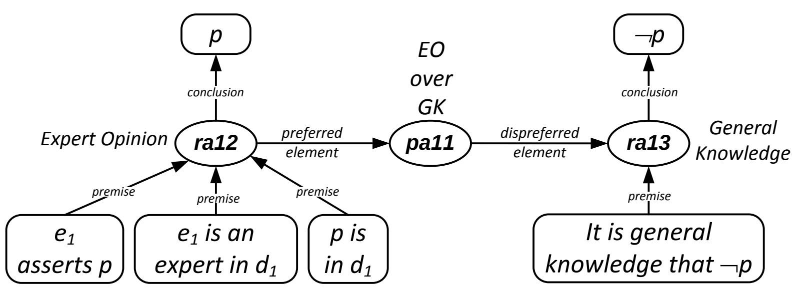

Figure 4: Preference between inferences

3 Abstract Argumentation and the ASPIC$^+$ Framework

The framework of [27] further develops the attempts of [1, 12] to integrate within Dung's abstract approach [14], the work of [26, 44, 28] on rule-based argumentation.

A Dung abstract argumentation framework assumes a given set of arguments and a binary attack relation between arguments, and then defines various ways to identify subsets of arguments ('extensions') that are in some sense acceptable. Dung abstracts from the structure of arguments and the nature of the attack relation, assuming that these are defined by some unspecified logical theory. Its level of abstraction, however, precludes giving guidance so as to ensure that the arguments of the instantiating theory that are identified as being acceptable, satisfy intuitively rational properties. Hence, the ASPIC framework of [1] adopted an intermediate level of abstraction, providing abstract accounts of the structure of arguments, the nature of attack, and the use of preferences. [12] then formulated consistency and closure postulates that cannot be formulated at Dung's fully abstract level, and showed these postulates to hold for a special case of ASPIC; one in which preferences were *not* accounted for. More recently, ASPIC$^+$ [27] generalised ASPIC and showed that the postulates were satisfied when applying preferences. The significance of this work is that ASPIC$^+$ captures a broad range of instantiating logics and argumentation systems, extending those captured by ASPIC (e.g., to additionally capture assumption-based argumentation [11] and systems using argument schemes). Furthermore, [23] has recently adapted ASPIC$^+$ to capture classical logic approaches to argumentation [6], including those that accommodate preferences [3, 2], and [41] have shown how arguments from Carneades [16] can be interpreted in ASPIC$^+$.

The ASPIC$^+$ framework as defined in [27] assumes an unspecified logical language and defines 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. Preferences may then be used to identify which attacks succeed as *defeats*, so that one obtains 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) or whether they express general patterns of inference, such as the deductive inferences of classical logic or defeasible argumentation schemes. The arguments and defeats defined by any logical formalism captured by the ASPIC$^+$ framework, then instantiate a Dung framework (DF), so that the acceptable arguments can then be evaluated.

Recently, [22] have extended ASPIC$^+$ so as to instantiate [20]'s extension of Dung's abstract approach. In [20], Modgil incorporates a second attack relation allowing for the possibility of *attacks on attacks* in addition to attacks on arguments. Intuitively, if argument C claims that argument B is preferred to argument A, and A attacks B, then C undermines the success of A's attack on B (i.e., A does not *defeat* B) by attacking A's attack on B. Since arguments attacking attacks can themselves be attacked, Modgil's *Extended Argumentation Frameworks* (*EAF*s) can fully model argumentation about whether one argument is preferred to another.

In the rest of this section, we first briefly review Dung's abstract approach and Modgil's extension of this approach. We then review [27]'s ASPIC$^+$ framework with fixed preferences and the modified version [22] that instantiates *EAF*s. For both versions, translations from and into the AIF will be defined in Section 4.

3.1 Abstract Argumentation Frameworks

A *Dung argumentation framework* (*DF*) [14] is a tuple $(\mathcal{A}, \mathcal{C})$, where $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation on the arguments $\mathcal{A}$. An argument $X \in \mathcal{A}$ is said to be acceptable w.r.t. some $S \subseteq A$ iff $\forall Y$ s.t. $(Y, X) \in \mathcal{C}$ implies $\exists Z \in S$ s.t. $(Z, Y) \in \mathcal{C}$. A *DF*'s characteristic function $\mathcal{F}$ is defined such that for any $S \subseteq \mathcal{A}$, $\mathcal{F}(S) = \{X | X$ is acceptable w.r.t. S $\}$. We now recall Dung's definition of extensions under different semantics:

Definition 3.1 Let $(\mathcal{A}, \mathcal{C})$ be a *DF*, $S \subseteq \mathcal{A}$ be *conflict free* (i.e., $\forall X, Y \in S$, $(X, Y) \notin \mathcal{C}$):
S is an *admissible* extension iff $S \subseteq \mathcal{F}(S)$; S is a *complete* extension iff $S = \mathcal{F}(S)$; S is a *preferred* extension iff S is a set inclusion maximal complete extension; S is a *grounded* extension iff S is a set inclusion minimal complete extension; S is a *stable* extension iff S is preferred and $\forall Y \notin S$, $\exists X \in S$ s.t. $(X, Y) \in \mathcal{C}$.
For $s \in \{$complete, preferred, grounded, stable$\}$, $X \in \mathcal{A}$ is *sceptically* justified under the s semantics, if X belongs to all s extensions, and *credulously* justified if X belongs to at least one s extension.

Extended Argumentation Frameworks (*EAF*s) [20] extend *DF*s to include a second attack (*pref-attack*) relation:

Definition 3.2 [*EAF*] An *EAF* is a tuple $(\mathcal{A}, \mathcal{C}, \mathcal{D})$, where $(\mathcal{A}, \mathcal{C})$ is a *DF*, $\mathcal{D} \subseteq \mathcal{A} \times \mathcal{C}$, and if $(Z, (X, Y))$, $(Z', (Y, X)) \in \mathcal{D}$ then (Z, Z'), $(Z', Z) \in \mathcal{C}$.

Note the constraint on any Z, Z', where given that they respectively pref-attack (X, Y) and (Y, X), then they express contradictory preferences (Y is preferred to X, respectively X is preferred to Y) and so themselves symmetrically attack each other. Modgil then defines modified notions of conflict free-ness and acceptability, and then, with the exception of the grounded semantics[5], defines extensions and justified arguments, as in Definition 3.1 (we refer the reader to [20] for the technical details).

3.2 ASPIC$^+$with fixed preferences

The basic notion of the ASPIC$^+$ framework is that of an argumentation system.

Definition 3.3 [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 $\mathcal{R}_d$ (where, as usual, $r_i < r_j$ denotes that $(r_i, r_j) \in\leq$, $(r_j, r_i) \notin\leq$).

Definition 3.4 [Logical language] Let $\mathcal{L}$, a set, be a logical language.

- φ is a *contrary* of ψ if $\varphi \in \overline{\psi}$, $\psi \notin \overline{\varphi}$;
- φ is a *contradictory* of ψ (denoted by '$\varphi = -\psi$'), if $\varphi \in \overline{\psi}$, $\psi \in \overline{\varphi}$.

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*, respectively *presumably*, the consequent φ holds'. As is usual in logic, inference rules can be specified by schemes in which a rule's antecedents and consequent contain metavariables ranging over $\mathcal{L}$ [15, 36]. For instance, in [27], the rule scheme $\{\varphi, \varphi \rightsquigarrow \psi \Rightarrow \psi$ (for all $\varphi, \psi \in \mathcal{L}$)$\}$ denotes the set of all Defeasible Modus Ponens inferences in $\mathcal{R}_d$.

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

Definition 3.5 [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 $\mathcal{K} \setminus K_n$ (where, as usual, $k_i <' k_j$ denotes that $(k_i, k_j) \in\leq'$, $(k_j, k_i) \notin\leq'$). 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.

[5] Since an *EAF*'s characteristic function is only monotonic for a special class of *hierarchical* frameworks, [20] defines the grounded extension of arbitrary *EAF*s as the fixed point obtained by iterative application (starting with $\emptyset$) of the characteristic function.

- $\mathcal{K}_p$ 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 [44], in which for any argument A, the function `Wff` returns all the formulas in A; `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 3.6 [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: $\texttt{Prem}(A) = \{\varphi\}$; $\texttt{Wff}(A) = \{\varphi\}$; $\texttt{Conc}(A) = \varphi$; $\texttt{Sub}(A) = \{\varphi\}$; $\texttt{Rules}(A) = \emptyset$; $\texttt{TopRule}(A)$ = 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 $\mathcal{R}_s/\mathcal{R}_d$.
$\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$,
$\texttt{Wff}(A) = \texttt{Wff}(A_1) \cup \ldots \cup \texttt{Wff}(A_n) \cup \{\psi\}$,
$\texttt{Conc}(A) = \psi$,
$\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$.
$\texttt{Rules}(A) = \texttt{Rules}(A_1) \cup \ldots \cup \texttt{Rules}(A_n) \cup \{\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow/\Rightarrow \psi\}$
$\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow/\Rightarrow \psi$

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 $\texttt{Prem}(A) \nsubseteq \mathcal{K}_n$.

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 that are not strict and firm, 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 defined in terms of the orderings on the elements of $\mathcal{R}_d$ and $\mathcal{K} \setminus K_n$ (given in Definitions 3.3 and 3.5). Because of space limitations we refer to [27] for two example definitions of argument orderings. The notion of an argument ordering is used in the notion of an argumentation theory.

Definition 3.7 [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 on 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 of 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 [27] leaves the nature of this naming convention implicit. In this paper we assume explicitly that this can be done in terms of a subset $\mathcal{L}_R$ of $\mathcal{L}$ containing formulas of the form r_i that denote the names of inference rules:

- $\mathcal{L}_R \subseteq \mathcal{L} = \{r_i \mid r_i \in \mathcal{R}\}$.

For convenience we will also use elements of $\mathcal{L}_R$ as names for inference rules at the metalevel, letting the context disambiguate. Furthermore, we make the assumption that for any AS, rules in $\mathcal{R}$ do not have their own name as an antecedent or consequent.

Definition 3.8 [Attacks] A *attacks* B iff A *undercuts*, *rebuts* or *undermines* B, where:

• Argument A *undercuts* argument B (on B') iff $\texttt{Conc}(A) \in \overline{r}$ [6] 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. Some kinds of attack succeed as *defeats* independently of preferences over arguments, whereas others succeed only if the attacked argument is not stronger than the attacking argument (see [24] for a detailed discussion of the rationale for this distinction).

Definition 3.9 [Defeat]
An undercut, contrary-rebut, or contrary-undermine attack is said to be *preference-independent*, otherwise an attack is *preference-dependent*.

Argument A *defeats* B iff A attacks B on B', and either A's attack on B is *preference-independent*, or; A's attack on B is *preference-dependent* and $A \not\prec B'$.

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 [27], structured argumentation theories are then linked to Dung-style abstract argumentation frameworks. Recall that such frameworks are a pair $\langle \mathcal{A}, \mathcal{C} \rangle$ where $\mathcal{A}$ is a set of arguments and $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{A}$. Then:

Definition 3.10 [*DF* corresponding to an AT] An *abstract argumentation framework DF_{AT} corresponding to an argumentation theory AT* is a pair $\langle \mathcal{A}, \mathcal{C} \rangle$ such that $\mathcal{A}$ is the set of arguments on the basis of AT as defined by Definition 3.6, and $\mathcal{C}$ is the defeat relation on $\mathcal{A}$ given by Definition 3.9.

Thus, any semantics for abstract argumentation frameworks can be applied to arguments in an ASPIC$^+$ framework. In [27] it is shown that for Dung's four original semantics [14], ASPIC$^+$frameworks as defined above satisfy [12]'s rationality postulates (if they satisfy some further basic assumptions).

3.3 E-ASPIC$^+$ with attacks on attacks

As described earlier, [20] incorporates a second attack relation allowing for the possibility of *attacks on attacks*. Furthermore, [22] had reason to refine Modgil's account with the possibility that an attack relation between arguments is attacked by a *set* of arguments. Thus an extended argumentation framework as defined in [22] is a tuple $(\mathcal{A}, \mathcal{C}, \mathcal{D})$, where $\mathcal{A}$ is a set of arguments, $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation between arguments, and $\mathcal{D} \subseteq (2^{\mathcal{A}}/\emptyset) \times \mathcal{C}$ contains the attacks on attacks, or "pref-attacks". Then for any set $S \subseteq \mathcal{A}$, we say that A *S-defeats* B iff (A, B) $\in \mathcal{C}$ and $\neg\exists\phi \subseteq S$ s.t. $(\phi,(A, B)) \in \mathcal{D}$. Thus defeat is made relative to a set of arguments; specifically what the conclusions of these arguments say about the relative preference of A and B. [20, 22] then defined argument extensions by adapting [14]'s definitions (in much the same way as in [20]'s original formalisation of *EAF*s). Since these definitions are irrelevant for present concerns, we shall not repeat them here.

[6] i.e., $\texttt{Conc}(A)$ is a contrary of the formula in $\mathcal{L}_R$ that names the rule r.

[22] then extended ASPIC$^+$ to E-ASPIC$^+$, which instantiates *EAF*s just as ASPIC$^+$ instantiates *DF*s. The attack relation $\mathcal{C}$ is defined as in ASPIC$^+$ with Definition 3.8. Then any reference in ASPIC$^+$ to the argument ordering $\preceq$ is removed, since this ordering is now the outcome of argumentation. For the same reason any reference to the partial preorders on the defeasible rules and knowledge base is removed. Instead a fully abstract partial function $\mathcal{P}$ is assumed that extracts orderings from sets of arguments that conclude preferences (over other arguments). These sets of preference arguments then collectively pref-attack attacks in order to undermine the success of the latter as defeats. Thus an extended argumentation theory is defined as follows:

Definition 3.11 [Extended Argumentation Theory]
– An *extended argumentation system* (EAS) is a triple $(\mathcal{L}, ^-, \mathcal{R})$.
– An *extended knowledge base* (EKB) is a set $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_p \cup \mathcal{K}_a$.
Let $\mathcal{A}$ denote the set of arguments based on an EKB $\mathcal{K}$ in an EAS $(\mathcal{L}, ^-, \mathcal{R})$, as defined in Definition 3.6.
– An *extended argumentation theory* is a triple EAT = $(EAS, EKB, \mathcal{P})$, where $\mathcal{P}$ is a partial function defined as:

$$\mathcal{P} : 2^{\mathcal{A}} \longrightarrow Pow(\mathcal{A} \times \mathcal{A}).$$

Henceforth we say that if $(X, Y) \in \mathcal{P}(\phi)$ then $Y \prec X \in \mathcal{P}(\phi)$.

That is to say, an extended argumentation theory now makes explicit reference to the arguments defined by the EKB, mapping sets of arguments to preference relations over individual arguments. This was done in order to remain as abstract as possible. So for example, [22] did not want to define instead a function that mapped from sentences in $\mathcal{L}$ to priority relations over pairs of sentences in $\mathcal{L}$ (e.g., $(r_1, r_2) \in \mathcal{P}(x \Rightarrow r_1 > r2)$) since this would compromise the generality of E-ASPIC$^+$ in that [22] wanted to make as minimal a commitment as possible as to how preferences are defined.

EATs are then linked to Extended Argumentation Frameworks as follows:

Definition 3.12 [*EAFC* for structured arguments] A *structured EAFC* corresponding to an EAT $(EAS, EKB, \mathcal{P})$, is a *EAFC* $(\mathcal{A}, \mathcal{C}, \mathcal{D})$ such that:

1. $\mathcal{A}$ is the set of arguments as defined in Definition 3.6;
2. $(A, B) \in \mathcal{C}$ iff A undercuts, rebuts or undermines B according to Definition 3.8;
3. $(\phi, (A, B)) \in \mathcal{D}$ iff $(A, B) \in \mathcal{C}$, and:
   (a) $\forall B' \in \texttt{Sub}(B)$ s.t. A rebuts or undermines B on B', $\exists \phi' \subseteq \phi$ s.t. $A \prec B' \in \mathcal{P}(\phi')$, and ϕ is a minimal (under set inclusion) set satisfying this condition; and
   (b) A does not preference-independent attack B.

We say that E is an extension of an EAT iff E is an extension of the structured *EAFC* corresponding to the EAT. Furthermore, we say that a structured *EAFC* $(\mathcal{A}, \mathcal{C}, \mathcal{D})$ is finite iff $\mathcal{A}$ is finite.

Definition 3.12 uses the $\mathcal{P}$ function, which needs to be defined separately. This function may depend on conclusions of arguments that represent preferences or priorities over premises or defeasible inference rules. In such a case, we assume a language $\mathcal{L}_m$ that allows expressions of the form $l > l'$, where l and l' are wffs in $\mathcal{L}$. We thus define for any extended argumentation system $EAS = (\mathcal{L}, ^-, \mathcal{R})$:

- $\mathcal{L}_m \subseteq \mathcal{L} = \{l > l' \mid l, l' \in \mathcal{L}\}$

Note that a preference between inference rules $r_i, r_j \in \mathcal{R}$ is expressed as a preference $r_i > r_j$ between the wffs that name the rules. Note also that this definition allows that $\mathcal{L}_m$ contains nested preference formulas. We further assume that any such EAS contains a set of strict rules PP axiomatising a partial preorder over $>$ (here, x, y, z are meta-variables ranging over rule names):

- $o_1 : (z > y) \wedge (y > x) \rightarrow (z > x)$ • $o_2 : (y > x) \rightarrow \neg(x > y)$

[22] give two definitions of the $\mathcal{P}$ function, capturing, respectively, the weakest- and last-link principle of [27]. The weakest-link principle prefers an argument A over an argument B if A is preferred to B on both their premises and their defeasible rules. In other words, $B \prec A \in \mathcal{P}(\phi)$ if there is a defeasible rule r in B such that for each defeasible rule r' in A, there is an argument in the set ϕ that has the conclusion $r' > r$. If both A and B have no defeasible rules, then $B \prec A \in \mathcal{P}(\phi)$, and if there is an ordinary or assumption premise l in B such that for each ordinary or assumption premise l' in A, there is an argument in the set ϕ that has a conclusion $l' > l$. The last-link principle works in much the same way, but considers not all defeasible rules but only the last defeasible rules and, moreover, looks at the premises only if the arguments are strict (i.e., contain no defeasible inference rules). That is, the principle prefers an argument A over another argument B if at least one of the last defeasible rules used in B is concluded to be of lower priority to every last defeasible rule in A, or, in case that both arguments are strict, if at least one ordinary or assumption premise in B s concluded to be of lower priority than every ordinary or assumption premise in A.

4 Translating the ASPIC$^+$ framework: fixed preferences

In this section the connection between the core AIF ontology (Section 2) and the ASPIC$^+$ argumentation framework (Section 3) will be defined. That is, it will be shown how an AIF argument graph can be interpreted in terms of the ASPIC$^+$ framework, and vice versa. Essentially, these translation functions allow us to specify AIF arguments in the ASPIC$^+$ framework. Thus, ASPIC$^+$ can then engage with other tools and methods in the AIF family.

As mentioned in Section 1, specifying the AIF ontology in a logical framework such as ASPIC$^+$ results in the ontology being given a formal grounding. There are few constraints on an argument expressed in the ontology's abstract language, as flexibility is needed if the AIF is to take into account the natural arguments put forth by people who do not always abide by strict formal rules of argument. However, the AIF also has a normative aim: to help people perform "good", i.e., rational, argument [32]; that is, to argue in a rational way. The ASPIC$^+$ framework sets rational boundaries for argumentation as well as providing for consistency checking and further evaluation of complex argument graphs.

A valid question is whether the boundaries set by the ASPIC$^+$ framework are the right ones; that is, does the ASPIC$^+$ framework provide for appropriate argumentation logics for expressing and evaluating arguments? We think that this is the case, mainly because the ASPIC$^+$ framework captures a broad range of existing argumentation formalisms from the literature. Moreover, the ASPIC$^+$ framework is embedded in Dung's approach [14] while, finally, under certain reasonable assumptions it satisfies the rationality postulates of [12]. Furthermore, in addition to the (E-)ASPIC$^+$ framework's obvious relation to [14, 26, 44, 28], several other well-known argumentation systems (e.g. [11],[6]) are shown by [27, 23] to be special cases of the (E-)ASPIC$^+$ framework. Thus, by connecting to ASPIC$^+$ we will have also shown the connection between the AIF and other logical languages and systems for argumentation.

In addition to formally grounding the AIF ontology, mapping the AIF onto the ASPIC$^+$ framework also tests the conceptual soundness of the ASPIC$^+$ framework; the AIF ontology, and particularly its hierarchy of argumentation schemes, is based on a tradition in philosophy and linguistics that has carefully examined the patterns and fallacies that occur in natural arguments [45]. Furthermore, with the AIF ontology as an interlingua, the expressiveness of ASPIC$^+$ can be compared to the expressiveness of other languages for argumentation (e.g. [16]'s framework) and thus the limits of ASPIC$^+$ may be analysed. Finally, there is the possibility of testing the ASPIC$^+$ framework by engaging with large corpora of natural argument that have been constructed in other tools that interface with the AIF (e.g. ArgDB [32]). Indeed, in Section 6, we will make use of the defined translations in order to examine some of the issues that may arise when interpreting AIF representations of diagrammed arguments in the ASPIC$^+$ framework.

When considering the translations proposed below, it should be noted that the AIF ontology is purely intended as a language for representing argumentative information consisting of propositions, inferences, conflicts and preferences. It is possible to run any number of (external) processes over this information to, for example, determine the number of premises of arguments, determine logical consequences of an argument, determine the acceptability of an argument, and so on. The properties thus calculated, the *calculated properties*, are dependent on non-argumentative processes (e.g. defining, counting, weighing, comparing) that are arbitrarily complex and arbitrarily specific. Hence, these processes themselves cannot be captured in the AIF ontology itself, and nor should they be, for otherwise the AIF would swell to become some unwieldy general purpose programming language. What this means is that using the AIF ontology as an interlingua can only ever guarantee representational isomorphism, that is, that the argumentative information and the argumentative relations between it are preserved. This is in contrast to the translations proposed by [41], which ensure *computational equipotence*, that is, that the same statements will be acceptable in both the Carneades and the ASPIC$^+$ framework.

We will illustrate the translation functions by means of an example AIF graph shown in Figure 5.

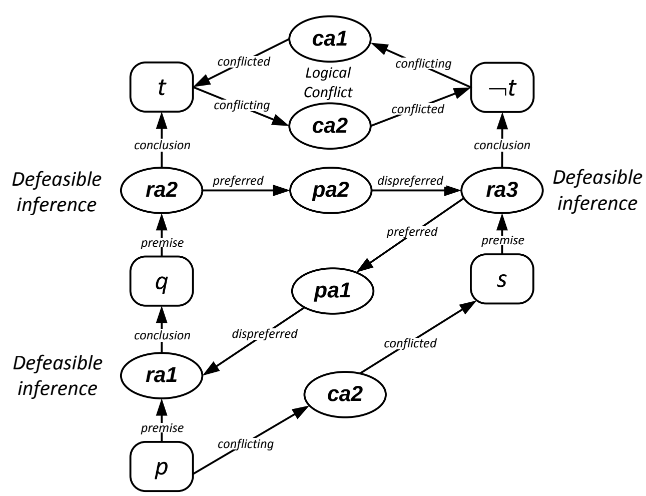

Figure 5: Example AIF graph

4.1 From the AIF ontology to the ASPIC$^+$ framework

We now define how an AIF argument graph can be interpreted in ASPIC$^+$. Since in ASPIC$^+$ the argumentation framework (Definition 3.10) is calculated from an argumentation theory (Definition 3.7), 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 complies with the philosophy underlying the AIF, which is a language for the representation of arguments and not for to the computation of (properties of) arguments. Properties such as defeat are thus *calculated properties* of an AIF graph, properties which can be inferred by some specific tool or framework that processes the graph.

Note that in this paper, we will leave the exact structure of the Forms ontology largely implicit and assume a set $\mathcal{F}$ that contains the relevant forms. In addition to the forms and edge types defined in Figure 1, we assume *deductive* and *defeasible* subclasses of inference schemes and classes of *ordinary premise*, *axiom* and *assumption* statements (subclasses of Statement Descriptions).

Definition 4.1 [Translating AIF to ASPIC$^+$]
Given an AIF argument graph G, a set of forms $\mathcal{F}$ and a set of fulfilment relations that link elements of G to elements of $\mathcal{F}$, an ASPIC$^+$ argumentation theory AT based on G is as follows:

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

The language of the argumentation theory consists of all I- and RA-nodes in the graph (1); notice that the inferences (RA-nodes) are translated as the subset $\mathcal{L}_R$ of $\mathcal{L}$. The knowledge base $\mathcal{K}$ consists of all initial nodes in the graph (2): nodes that explicitly fulfil a form *axiom*[7] or *assumption* become members of their respective subsets of $\mathcal{K}$, all other nodes are considered to be 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 (3). The type of inference rule is determined by the form that the RA-node uses. Note that here in fact we translate *applications* of inference schemes from $\mathcal{F}$ as inference rules in $\mathcal{R}$. Strictly speaking, RA-nodes should be translated as $\texttt{Rules}(A_1) \cup \ldots \cup \texttt{Rules}(A_n)$, where $\mathcal{A}_{AT} = \{A_1, \ldots, A_n\}$ is the set of arguments that follow from the argumentation theory AT. However, since this set of arguments is not directly translated from the AIF graph (only the information that ASPIC$^+$ needs in order to infer or "calculate" these arguments), we cannot define the translation of RA-nodes thus. This is in practice not a problem, as in the ASPIC$^+$ framework, any relevant inference rule is automatically applied. That is, if there is an inference rule $p \rightarrow q \in \mathcal{R}$ and $p \in \mathcal{K}$, there will be an argument in $\mathcal{A}_{AT}$ in which this rule is applied with premise p and conclusion q. Furthermore,

[7] Notice that here we allow for an additional subclass *axiom* of the *Statement Description* class in our AIF ontology (Figure 1).

we leave the exact translation of schemes in the Forms Ontology to rule schemes in the ASPIC$^+$ framework implicit for now, as this would involve a more specific definition of rule schemes in the ASPIC$^+$ theory.

Contrariness is determined by whether two nodes are connected through a CA-node (4). Finally, a PA-node between two initial I-nodes or between two RA-nodes translates into preferences between either elements of $\mathcal{K}$ ($\leq'$) or between inference rules ($\leq$), respectively (5, 6). Now, the ASPIC$^+$argumentation theory based on the AIF graph in Figure 5 and its forms, is as follows.

- $\mathcal{L} = \{p, q, s, t, \neg t, r_1, r_2, r_3\}$;
- $\mathcal{K}_n = \emptyset; \mathcal{K}_p = \{p, s\}; \mathcal{K}_a = \emptyset$;
- $\mathcal{R}_s = \emptyset; \mathcal{R}_d = \{r_1 : p \rightarrow q, r_2 : q \Rightarrow t, r_3 : s \Rightarrow \neg t\}$;
- $\overline{s} = \{p\}; \overline{\neg t} = \{t\}; \overline{t} = \{\neg t\}$
- $\leq' = \emptyset; \leq = \{r_1 < r_3, r_3 < r_2\}$.

Given this argumentation theory, the following arguments can be built:

$$\begin{array}{lll} A_1 : p & A_2 : A_1 \Rightarrow_{r1} q & A_3 : A_2 \Rightarrow_{r2} t \\ B_1 : s & B_2 : B_1 \Rightarrow_{r3} \neg t & \end{array}$$

One difference between an AIF argument graph and its translation into the ASPIC$^+$ framework is that the preferences in the original AIF graph do not necessarily obey the constraints of a partial preorder, as do $\leq$ and $\leq'$, since users of the AIF are essentially free to ignore these constraints. If, for example, we would try to translate an AIF graph which expresses a non-transitive ordering of preferences, this will yield an error as such an ordering is not possible in the ASPIC$^+$ framework. This illustrates how ASPIC$^+$ sets rational boundaries for argumentation.

In other cases, however, it might not be possible to translate a certain AIF graph to ASPIC$^+$ because of a limitation of the ASPIC$^+$ framework. For example, in the AIF, it is possible to give reasons for and against contrariness and preferences (e.g. by supporting PA- or CA-nodes with an I-node through an RA-node). In the ASPIC$^+$ framework reasons for or against preferences and contrariness relations cannot be given[8] and the translation function hence does not allow for such constructions to be translated (e.g. a link between an RA-node and a PA-node is left untranslated even when the PA-node is the conclusion of the inference denoted by the RA-node.

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. Hence, for any function f defined on arguments (in Definition 3.6), we overload the symbol f to let, for any set $S = \{A_1, \ldots, A_n\}$ of arguments, $f(S)$ stand for $f(A_1) \cup \ldots \cup f(A_n)$. Furthermore, as with the translation from AIF to ASPIC$^+$ we will leave the translation of rule schemes in ASPIC$^+$ to schemes in $\mathcal{F}$ in the AIF implicit. This means that all inferences, preferences and conflicts in the original ASPIC$^+$ framework are applied (as non-applied inferences, preferences and conflicts are represented by schemes). What this effectively entails is that we translate only the contrariness function and the preference relations $\leq'$ and $\leq$ that hold between rules which are actually used in an argument that is part of the argumentation framework.

[8]The E-ASPIC$^+$ framework does allow for reasoning about preferences, see sections 3.3 and 5.

Definition 4.2 [Translating ASPIC$^+$ to AIF]
Let $Args_{AT}$ be the set $\mathcal{A}$ of arguments on the basis of an ASPIC$^+$ argumentation theory AT. Then an AIF graph G, a set of forms $\mathcal{F}$ and a set of fulfilment relations that link elements of G to elements of $\mathcal{F}$, on the basis of $Args_{AT}$, is as follows:

1. I is the set of all distinct nodes v such that:
   (a) $v \in \texttt{Wff}(\mathcal{A}) \setminus \mathcal{L}_R$;
   (b) if $v \in \mathcal{K}_{n/p/a}$ then v fulfils a form *axiom/ordinary premise/assumption* $\in \mathcal{F}$.
2. RA is the set of all distinct nodes r for each rule named r in $\texttt{Rules}(\mathcal{A})$, where if $r \in \mathcal{R}_{s/d}$ then r fulfils a *deductive inference scheme/defeasible inference scheme* $\in \mathcal{F}$.
3. CA is the set of all distinct nodes v for each pair $\varphi, \psi \in \texttt{Wff}(\mathcal{A})$ and $\varphi \in \overline{\psi}$ (we say that v corresponds to (φ, ψ));
4. PA is the set of all 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. Given (1) – (4), E is the set such that for all v, v' in $I \cup RA \cup PA \cup CA$:
   (a) If $v \in I \cup RA$ and $v' \in RA$ then:
      i. $e = (v, v') \in E$ and e is of type *premise* if v is an antecedent of v';
      ii. $e = (v', v) \in E$ and e is of type *conclusion* if v is the consequent of v';
   (b) If $v \in I \cup RA$ and $v' \in CA$ and v' corresponds to (φ, ψ), then:
      i. $e = (v, v') \in E$ and e is of type *conflicting element* if $v = \varphi$;
      ii. $e = (v', v) \in E$ and e is of type *conflicted element* if $v = \psi$.
   (c) If $v \in I \cup RA$ and $v' \in PA$ and v' corresponds to (φ, ψ), then:
      i. $e = (v, v') \in E$ and e is of type *preferred element* if $v = \varphi$;
      ii. $e = (v', v) \in E$ and e is of type *dispreferred element* 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 formulas in an argument in $\mathcal{A}$, and where appropriate, forms in $\mathcal{F}$ are associated with these I-nodes (1). In our example (Figure 5), p, q, s, t and $\neg t$ are thus I-nodes, and p and s are of the form *premise*. 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 (2). In Figure 5, the nodes ra_1, ra_2 and ra_3 are based on r_1, r_2 and r_3, respectively, and all these RA-nodes fulfil *defeasible inference schemes* $\in \mathcal{F}$. CA-nodes correspond to conflicts between formulas occurring in arguments in $\mathcal{A}$ as determined by the contrariness relation (3). The nodes ca_1 and ca_2 are based on the contrariness between t and $\neg t$, and ca_3 is based on $\overline{s} = \{p\}$. Finally, in (4), PA-nodes correspond to the preferences in AT between the rules used in arguments in $\mathcal{A}$ ($\leq$) or between the premises of arguments in $\mathcal{A}$ ($\leq'$). In the example, there are two preferences $r_1 \leq r_3, r_3 \leq r_2$ which translate into pa_1 and pa_2, respectively. 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 (for example, the edges from p to ra_1 to q in Figure 5). Reasons for inference rules can be appropriately translated as links from RA-nodes to RA-nodes: condition 5a 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 claiming 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 ra_3 to pa_1 to ra_1 are based on $r_1 \leq r_3$ (i.e. pa_1 corresponds to $(r_1, r_3) \in \leq$). An undercutter can be expressed as a link from the conclusion of the undercutting argument, an I-node, to a CA-node and a link from this CA-node to the RA-node denoting the undercut rule.

4.3 Identity-preserving translations for ASPIC$^+$

Ideally, translating from AIF to some (formal) language and back again yields the original AIF graph. Whether a set of arguments expressed in ASPIC$^+$ or AIF are representationally isomorphic depends on the expressiveness of both the AIF language and ASPIC$^+$. As was discussed in the previous sections, there are some AIF structures that cannot be expressed in ASPIC$^+$. However, we can prove that the translation functions are identity-preserving (i.e. translating from AIF to ASPIC$^+$ and back again yields the same graph as we started out with) if we enforce some assumptions on the original graph. The conditions (3) – (6) on the graph set out in Definition 2.1 apply, and they ensure that the S-nodes always have their required predecessors and successors (i.e. RA-nodes have at least one *premise* and exactly one *conclusion*, PA-nodes have exactly one *preferred element* and exactly one *dispreferred element* and CA-nodes have exactly one *conflicting element* and exactly one *conflicted element*). In addition, we make some further assumptions on the graph so that it does not represent structures that cannot be handled by ASPIC$^+$. In particular, normal ASPIC$^+$ does not allow us to talk *about* preferences or contrariness relations in the object language and hence, PA- or CA-nodes cannot be the premises or conclusions of RA-nodes, the preferred element or dispreferred element of a PA-node or the conflicting element or conflicted element of a CA-node.

Assumption 4.3 In no AIF-graph is a PA- or CA-node connected to an RA-node via a *premise* or *conclusion* edge, or connected to a PA-node via a *preferred element* or *dispreferred element* edge, or connected to a CA-node via a *conflicting element* or *conflicted element* edge.

Under these conditions and assumptions, it can be proven that all translations from the AIF to ASPIC$^+$ and then back, result in an AIF graph that is isomorphic with the original graph in that the graphs differ at most in their names for the nodes.

Theorem 4.4 Let G' be an AIF graph satisfying Assumption 4.3, and AT be the ASPIC$^+$ argumentation theory based on G'. Let G be an AIF graph satisfying Assumption 4.3 based on $Args_{AT}$. Then G is isomorphic to G'.

The formal proofs can be found in the appendix. Note that here, we only prove that translating an AIF graph to ASPIC$^+$ and back again yields the same graph. These formal results do not mean that translating an ASPIC$^+$ set of arguments to AIF and back again yields the same set of arguments. The reason for this is that in this paper the AIF ontology is used as the interlingua and we consider ASPIC$^+$ as a specification of this abstract interlingua. Were we to use ASPIC$^+$ as an interlingua in the same way we use the AIF ontology, we would then need to prove that the translation from ASPIC$^+$ to AIF and back to ASPIC$^+$ is also identity-preserving. This proof would be more or less analogous to the proof of Theorem 4.4. Furthermore, in the same way that Theorem 4.4 assumes a particular AIF graph, we would need to enforce some assumptions on the ASPIC$^+$ theory so that theories that the AIF cannot express are not considered (e.g. only ASPIC$^+$ theories in which all rules in $\mathcal{R}$ are used in the construction of arguments, should be considered, otherwise the graph will have unconnected RA-nodes, which is not possible in the AIF).

5 Translating the E-ASPIC$^+$ framework: defeasible preferences

5.1 From the AIF ontology to the E-ASPIC$^+$ framework

We next present a translation from the AIF to E-ASPIC$^+$. We now want to replace clauses (5) and (6) of Definition 4.1 with a translation of AIF preference structures into an E-ASPIC$^+$ argumentation theory. Such a theory gives rise to a set of E-ASPIC$^+$ arguments, from which E-ASPIC$^+$'s $\mathcal{P}$ function then extracts an argument ordering. Since in general the $\mathcal{P}$ function is undefined, we cannot give a general translation of an AIF graph to the input needed by the $\mathcal{P}$ function; all we can give is translations for specific kinds of $\mathcal{P}$ functions. Therefore, we give a translation for $\mathcal{P}$ functions that define the argument ordering in terms of orderings of the defeasible rules and the non-axiom premises (cf. Section 3.3) We hence assume the language $\mathcal{L}_m \subseteq \mathcal{L}$ that allows us to express preference predicates (denoted as $\varphi > \psi$) and that R_s contains the set of rules $PP = \{o_1, o_2\}$ (in Section 3.3) that axiomatise the partial preorder over $>$.

Our new translation definition thus allows for formulas of the form $l > l'$, where l and l' are terms denoting I-nodes, RA-nodes and PA-nodes. This new translation must extract preference statements from I $\longrightarrow$ PA $\longrightarrow$ I, RA $\longrightarrow$ PA $\longrightarrow$ RA, PA $\longrightarrow$ PA $\longrightarrow$ PA structures in an AIF graph. Thus, we relax the restrictions on the AIF graph that were mentioned in section 4.3. Note that CA $\longrightarrow$ PA $\longrightarrow$ CA structures still cannot be translated since E-ASPIC$^+$ does not allow for preferences between contrariness relations. Furthermore, the translations of inference rules and contrariness relations must also be amended: Suppose in G an RA-node *ra1* instantiating a deductive scheme has a set S of I-nodes as predecessors and a PA-node pa as successor, where pa connects I-node p to I-node q. Clause (3) of the old translation function (in Definition 4.1) translates this into a strict rule $S \rightarrow pa$, but we want instead the rule $S \rightarrow p > q$.

Another issue for the new translation function is how to disambiguate the multiple incoming links into a PA-node in a situation where a reason for a preference has been given. In such a situation, we cannot simply say that the predecessor of a PA-node v is preferred to its successor v' (i.e. $v > v'$) since an (RA-node) predecessor of a PA-node may denote the inference for the preference statement expressed by the PA-node, as its conclusion. For example, in Figure 6, *ra1* is a predecessor of *pa1* but *ra1* is not preferred to q; rather, p is preferred to q and *ra1* denotes the inference (from u) to the preference expressed in *pa1* ($q \leq p$). In order to correctly translate a graph such as this, we have to use the Forms Ontology to realize the disambiguation. We thus assume that preferred and dispreferred nodes in a graph fulfil the appropriate forms in $\mathcal{F}$.

This leads to the following translation function from the AIF into E-ASPIC$^+$:

Definition 5.1 [Translating AIF to E-ASPIC$^+$]
Given an AIF argument graph G, a set of forms $\mathcal{F}$ and a set of fulfilment relations that link elements of G to elements of $\mathcal{F}$, an E-ASPIC$^+$argumentation theory EAT based on G is as follows (we let formulas be denoted by the same kind of terms, letting the context disambiguate).

1. $\mathcal{L} = \mathcal{L}_o \cup \mathcal{L}_R \cup \mathcal{L}_m$ where:
   (a) $\mathcal{L}_o = I$, $\mathcal{L}_R = RA$;
   (b) $\mathcal{L}_m$ is recursively defined as $\mathcal{L}_m^0 \cup \ldots \cup \mathcal{L}_m^n$ where:
      - $\mathcal{L}_m^0 = \{v_i > v_j \mid v_i$ and v_j are both in I or both in RA; v_i and v_j are the direct predecessor and direct successor of a PA-node $pa \in PA$ via a *preferred* and *dispreferred* edge, respectively (we say that $v_i > v_j$ is based on pa).
      - $\mathcal{L}_m^n = \mathcal{L}_m^{n-1} \cup \{\varphi > \psi) \mid$
        i. $\varphi, \psi \in \mathcal{L}_m^{n-1}$; and
        ii. φ, ψ are based on $v_i, v_j \in I \cup RA \cup PA$; and

iii. v_i and v_j are the direct predecessor and direct successor of a PA-node pa in G via a *preferred* and *dispreferred* edge, respectively (we say that $\varphi > \psi$ is based on pa).

2. $\mathcal{K} = \{v \mid v \in I$ and v is an initial node$\} \cup \{\varphi \in \mathcal{L}_m \mid \varphi$ is based on $v' \in PA$ where v' has at most one direct predecessor$\}$, where
   (a) $v \in \mathcal{K}_{n/a}$ if v fulfils a form *axiom/assumption*; and
   (b) $v \in \mathcal{K}_p$ otherwise.

3. $\mathcal{R}_s$/$\mathcal{R}_d$ is the set of all inference rules $\varphi_1, \ldots, \varphi_{n-1} \rightarrow/\Rightarrow \varphi_n$ for which there is a node $v_k \in RA$ such that:
   (a) v_k fulfils a *deductive/defeasible scheme* $\in \mathcal{F}$; and
   (b) v_k's direct predecessors connected to v_k via a *premise* edge are $v_1, \ldots, v_{n-1}$ and v_k's direct successor connected to v_k via a *conclusion* edge is v_n such that for all $1 \leq i \leq n$: $\varphi_i = v_i$ or φ_i is based on v_i.

4. $\varphi_i \in \overline{\varphi_j}$ iff there is a node $v_k \in CA$ such that v_k has a direct predecessor v_i and a direct successor v_j such that $\varphi_i = v_i$ or φ_i is based on v_i and $\varphi_j = v_j$ or φ_j is based on v_j.

The language of the argumentation theory (1) is expanded to include preference statements. Notice that in the definition of $\mathcal{L}_m$, we need the Forms Ontology to correctly determine whether a predecessor of a PA-node is the preferred element corresponding to the preference application. The translation further needs to keep track of the connection between an E-ASPIC$^+$preference statement and the PA-node it was based on, so that conflicts between preferences and reasons for preferences can be correctly translated.

The translation to the knowledge base (2) must also be amended: in addition to the propositions that correspond to initial nodes, $\mathcal{K}$ also contains preference statements based on PA-nodes that have not been inferred from some other information, but are themselves premises. This is ensured by allowing only PA-nodes that have at most one direct predecessor. To see why, observe that a preference of node v over node v' is at present defined in the AIF as $v \longrightarrow PA \longrightarrow v'$. Hence, if a PA-node has more than one predecessor, one of those predecessors must be the RA-node that denotes the inference for which the PA-node is the conclusion.

Finally, as was already mentioned just prior to Definition 5.1, the translations of inference rules (3) and conflict (4) have to be slightly amended so that reasons for preference statements and conflict between preference statements are correctly extracted from the graph.

As for the ASPIC$^+$ framework with fixed preferences, E-ASPIC$^+$ sets some rational boundaries for argumentation. For example, the preferences in the E-ASPIC$^+$ framework obey the constraints of a partial preorder. Furthermore, the translation function ensures that only preferences between two nodes of the same type (I-nodes or RA-nodes) are incorporated into the logical framework. However, there are still some limitations on the E-ASPIC$^+$ framework: for example, reasons for conflict relations or preferences between conflicts cannot be expressed in E-ASPIC$^+$.

In order to illustrate the above translation function we consider an example that is slightly more complex than the one in Figure 5. This new example (Figure 6) illustrates the use of preference arguments to resolve conflicts. It also illustrates that not all preferences that can be stated in an AIF graph are used by E-ASPIC$^+$ to compute defeats, even though they are translated. Let us say that all RA-nodes fulfil a defeasible inference scheme. Now, the E-ASPIC$^+$ theory EAS based on the graph is as follows:

- $\mathcal{L}_o = \{p, q, s, t, u, r_1, r_2, r_3\}$;
- $\mathcal{L}_m = \{p > q, q > p, r_3 > r_2, s > t\}$;

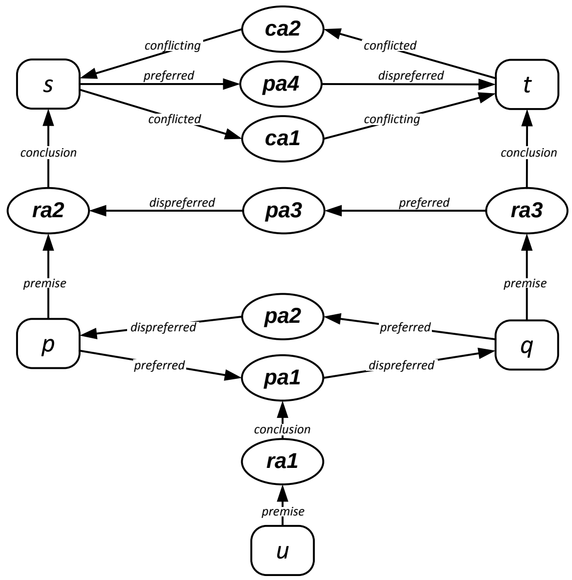

Figure 6: Preferences over arguments

- $\mathcal{K}_n = \emptyset; \mathcal{K}_p = \{p, q, u, q > p, r_3 > r_2, s > t\}; \mathcal{K}_a = \emptyset$
- $\mathcal{R}_s = PP; R_d = \{r_1 : u \Rightarrow p > q, r_2 : p \Rightarrow s, r_3 : q \Rightarrow t\}$;
- $\overline{s} = \{t\}, \overline{t} = \{s\}$.

Given this argumentation theory, the following arguments can be built. The arguments that can be built on the basis of the rules PP (which axiomatise a partial preorder) are not shown.

$A_1 : p$	$A_2 : A_1 \Rightarrow_{r2} s$
$B_1 : q$	$B_2 : B_1 \Rightarrow_{r3} t$
$C_1 : u$	$C_2 : C_1 \Rightarrow_{r1} p > q$
$D_1 : q > p$	
$E_1 : r_3 > r_2$	
$F_1 : s > t$	

Given the above set of arguments, Definition 3.8 determines that A_2 attacks B_2 and vice versa. Which one of these arguments ultimately defeats the other depends on whether we choose to adhere to the last-link or the weakest-link principle. In the case of the last-link principle, argument E_1, which expresses a preference of a rule in B_2 over a rule in A_2, attacks the attack from A_2 to B_2 and thus ensures that B_2 defeats A_2. According to the weakest-link ordering, both arguments are defensible: the defeasible rules in B_2 are still stronger than those in A_2, but the elements p and q in $\mathcal{K}$ are equally strong. Note that not all preferences have been used here: the preference between t and s is a premise of E-ASPIC$^+$, but it is easy to verify that neither the last- nor the weakest-link ordering use this preference to determine defeat (since these orderings only make use of priorities over premises and defeasible rules).

In order to illustrate the recursive definition of $\mathcal{L}_m$, consider the AIF graph in Figure 7. This graph leads to the following contents of $\mathcal{L}$.

- $\mathcal{L}_o = \{p, q\}$;
- $\mathcal{L}_m = \mathcal{L}_m^0 \cup \mathcal{L}_m^1$ where

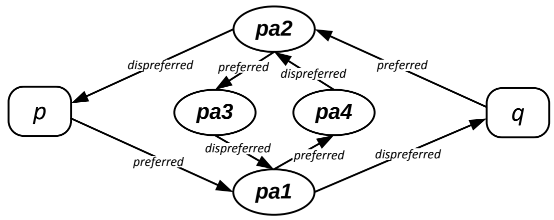

Figure 7: Preferences on preferences

– $\mathcal{L}_m^0 = \{p > q, q > p\}$;
– $\mathcal{L}_m^1 = \{(q > p) > (p > q), (p > q) > (q > p)\}$;

Note that in $\mathcal{L}_m$ we have that $p > q$ is based on pa_1, $q > p$ is based on pa_2, $(q > p) > (p > q)$ is based on pa_3 and $(p > q) > (q > p)$ is based on pa_4.

5.2 From the E-ASPIC$^+$ framework to the AIF ontology

We now turn to the translation from E-ASPIC$^+$ to AIF. Just as for ASPIC$^+$, we give the translation for a given set of arguments $\mathcal{A}$, and just as for the translation of the AIF to E-ASPIC+, we only give the translation for $\mathcal{P}$ functions that define the argument ordering in terms of orderings of the defeasible rules and the non-axiom premises. Furthermore, as with the translation from ASPIC$^+$ to AIF we assume that all preferences are applied, that is, there is no preference statement $\varphi > \psi \in \mathcal{L}_m$ for which either φ or ψ is not part of an argument in $\mathcal{A}$ (i.e. both φ and ψ are in $\texttt{Rules}(\mathcal{A})$ or in $\texttt{Wff}(\mathcal{A})$).

Definition 5.2 [Translating E-ASPIC$^+$ to AIF]
Let $Args_{EAT}$ be the set $\mathcal{A}$ of arguments on the basis of an E-ASPIC$^+$ argumentation theory EAT. Then an AIF graph G, a set of forms $\mathcal{F}$ and a set of fulfilment relations that link elements of G to elements of $\mathcal{F}$, on the basis of $Args_{EAT}$, is as follows:

1. I is the set of all distinct nodes v such that:
   (a) $v \in \texttt{Wff}(\mathcal{A}) \setminus (\mathcal{L}_R \cup \mathcal{L}_m)$
   (b) if $v \in \mathcal{K}_{n/p/a}$ then v fulfils a form *axiom/ordinary premise/assumption* $\in \mathcal{F}$.
2. RA is the set of all distinct nodes r for each rule named r in $\texttt{Rules}(\mathcal{A})$, where if $r \in \mathcal{R}_{s/d}$ then r fulfils a *deductive inference scheme/defeasible inference scheme* $\in \mathcal{F}$, respectively.
3. CA is the set of all distinct nodes v for each pair $\varphi, \psi \in \texttt{Wff}(\mathcal{A})$ and $\varphi \in \overline{\psi}$ (we say that v corresponds to (φ, ψ));
4. PA is the set of all distinct nodes v for each wff $\varphi > \psi \in \mathcal{L}_m \cap \texttt{Wff}(\mathcal{A})$ (we say that v corresponds to (φ, ψ));
5. E is the set such that for all v, v' in G:
   (a) If $v \in I \cup RA \cup PA$ and $v' \in RA$ then:
      i. $e = (v, v') \in E$ and e is of type *premise* if v is an antecedent of v';
      ii. $e = (v', v) \in E$ and e is of type *conclusion* if v is the consequent of v';
   (b) If $v \in I \cup RA \cup PA$ and $v' \in CA$ and v' corresponds to (φ, ψ), then:
      i. $e = (v, v') \in E$ and e is of type *conflicting element* if $v = \varphi$;
      ii. $e = (v', v) \in E$ and e is of type *conflicted element* if $v = \psi$.

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

i. $e = (v', v) \in E$ and e is of type *premise* if $\varphi > \psi$ is the antecedent of v;

ii. $e = (v, v') \in E$ and e is of type *conclusion* if $\varphi > \psi$ is the consequent of v;

iii. $e = (v, v') \in E$ and e is of type *preferred element* if $v = \varphi$;

iv. $e = (v', v) \in E$ and e is of type *dispreferred element* if $v = \psi$.

The set of I-nodes (1) is similar to that in Definition 4.2, except that here the preference statements are not translated as I-nodes (but rather as PA-nodes). In the example (Figure 6), p, q, s, t, u are I-nodes. The translations of RA-nodes (2) and CA-nodes (3) are the same as in the original ASPIC$^+$ translation (Definition 4.2). Thus, in Figure 6, the nodes ra_1, ra_2 and ra_3 are based on r_1, r_2 and r_3, respectively, and all these RA-nodes fulfil *defeasible inference schemes* $\in \mathcal{F}$. The nodes ca_1 and ca_2 are based on the conflict between t and s.

PA-nodes are of course defined differently, as they are based on preference statements rather than some predefined set of preferences. These preference statements in an EAT are between the rules used in arguments in $\mathcal{A}$ ($\leq$) or between the premises of arguments in $\mathcal{A}$ ($\leq'$). In the example, there are four preferences $p > q, q > p, r_3 > r_2, s > t$ which translate into pa_1, pa_2, $pa3$ and $pa4$, respectively. Since the argument ordering $\preceq$ of EAT is defined in terms of $\leq$ and $\leq'$, it is not an explicit additional part of the AIF graph.

The edges between the nodes are again determined in terms of the relations between the corresponding elements in the EAT. For RA-nodes and CA-nodes (5a and 5b) the definition is the same as Definition 4.2. For PA-nodes (5c), however, there is a slight difference: there is an edge from a PA-node to an RA-node if the RA-node denotes an inference rule that has the preference statement denoted by the PA-node as an antecedent(5c(i)), and there is an edge from an RA-node to a PA-node if the RA-node denotes the inference rule that is used to infer the preference statement denoted by the PA-node (5c(i)).

When preferences in ASPIC$^+$ are translated to AIF, the preferences in the graph will adhere to the conditions of a partial preorder but these conditions will not be explicit. In E-ASPIC$^+$, these conditions are made explicit as strict inference rules (the strict rules PP in Section 3.3) so they are translated to the AIF graph. Take, for example, transitivity. Assume the following argumentation theory:

- $\mathcal{L}_o = \{p, q, r, r_1\}$;
- $\mathcal{L}_m = \{p > q, q > r, p > r\}$;
- $\mathcal{K}_n = \emptyset$; $\mathcal{K}_p = \{p > q, q > r\}$; $\mathcal{K}_a = \emptyset$
- $\mathcal{R}_s = \{\mathit{Transitivity} : p > q, q > r \rightarrow p > r\}$ and $R_d = \emptyset$;
- There are no contrariness relations.

Note here that the $\mathit{Transitivity}$ rule is one of the rules in PP that was implicitly assumed to be part of every E-ASPIC$^+$ argumentation theory. Now, given this argumentation theory the following arguments can be built:

$$A_1 : p > q \quad A_2 : q > r \quad A_3 : A_1, A_2 \rightarrow_{r1} p > r$$

These arguments can then be translated to AIF, which gives us the graph in Figure 8. Notice that q fulfils both the form *preferred* and the form *dispreferred*, but for distinct preferences. The current translation functions do not keep track of the particular preference for which an element is either preferred or dispreferred as this is not necessary for a correct translation.

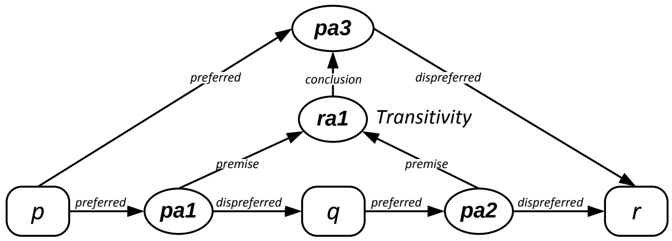

Figure 8: Deriving new preferences from old ones with transitivity

5.3 Identity-preserving translations for E-ASPIC$^+$

Just as for ASPIC$^+$, translating from AIF to E-ASPIC$^+$ and back to AIF yields the same graph as the original one. However, the conditions on the graph for E-ASPIC$^+$ are slightly different, because of the possibility of reasons for preferences between, and conflicts between, preferences. Hence, we need only assume that CA-nodes cannot be the premises or conclusions of RA-nodes, the preferred element or dispreferred element of a PA-node or the conflicting element or conflicted element of a CA-node.

Assumption 5.3 In no AIF-graph is a CA-node connected to an RA-node via a *premise* or *conclusion* edge, or connected to a PA-node via a *preferred element* or *dispreferred element* edge, or connected to a CA-node via a *conflicting element* or *conflicted element* edge.

Theorem 5.4 Let G' be an AIF graph satisfying Assumption 5.3, and EAT be the E-ASPIC$^+$ argumentation theory based on G'. Let G be an AIF graph satisfying 5.3 based on $Args_{EAT}$. Then G is isomorphic to G'.

6 Evaluating Diagrammed Arguments via the AIF to ASPIC$^+$ Translation

As stated in Section 1, one of the AIF's main practical goals has been to facilitate the research and development of various tools for argument manipulation and visualization. In particular, [13] envisaged the use of the AIF as an interlingua linking the aforementioned tools to software components for evaluating the dialectical status of arguments under the various semantics proposed for Dung's abstract argumentation frameworks [14]. The work presented in the previous sections represents an important step towards realising this use. Given an AIF translation of diagrammed arguments, one can then employ Definition 4.1's translation to obtain the corresponding ASPIC$^+$ argumentation theory. The abstract argumentation framework corresponding to the argumentation theory can then be obtained as defined in Definition 3.10, so that the status of the diagrammed arguments can then be evaluated under the various semantics.

In order to realise this use of the AIF and its ASPIC$^+$ translation, the specific formats that are output by the diagramming tools will have to be translated to the language of the AIF ontology. This translation includes both a theoretical and a practical component. On a theoretical level, we define translations between abstract, high-level renderings of the AIF ontology (as proposed in this paper) and the appropriate data model of the individual diagramming tool. On a more practical level, the actual file format of the diagramming tool will have to be translated into an implementation of the AIF ontology. Currently, the AIF ontology is implemented as OWL/XML, RDF/XML and an SQL database. This database, the AIFdb [18][9] allows for the storage and retrieval of AIF compliant argument structures. AIFdb offers a rich array of web

[9] http://www.arg.dundee.ac.uk/AIFdb/

service interfaces allowing for interaction with low level argument components (nodes, edges, schemes), as well as modules which handle the import and export of numerous formats such as SVG, DOT, RDF/XML and the formats of the Carneades, Rationale and Araucaria tools. To facilitate import and export the database also allows for the creation of 'Argument Maps', sets of nodes which can then be used to perform operations on a specific grouping of argument components. In this section, we explore some of the issues behind the translation of one of these formats, namely that of the Rationale tool.

The Rationale tool [5] has been developed for nurturing critical thinking skills by allowing users to organize information, visualize argumentation, and subsequently build well-founded arguments, and to identify, analyze and evaluate argumentation presented by others. It visualises information (i.e. claims, sentences, propositions) as text boxes and includes two main relations between these pieces of information: (supporting) *reasons* and (opposing) *objections*. Consider the Rationale diagramming of the argument in Figure 9, which illustrates a user's diagramming of a statement $i2$: 'It is sunny today' supporting (green) the claim that $i1$: 'I should go to the beach today', and a statement $i3$: 'The surf is dangerous today' opposing (red) the claim that 'I should go to the beach today'.

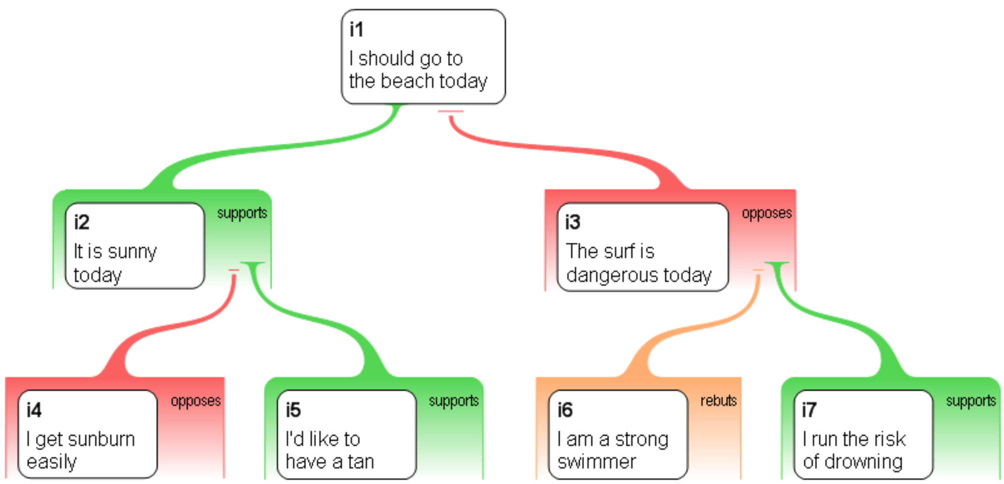

Figure 9: An argument in Rationale

Rationale diagrams can be translated to AIF graphs in the following way.

Definition 6.1 [Translating Rationale to AIF] Given a Rationale diagram D, an AIF graph G on the basis of D is as follows.

1. I is the smallest set consisting of distinct nodes v for each text box in the diagram.
2. RA is the smallest set consisting of distinct nodes v for each (green) reason link in the diagram.
3. CA is the smallest set consisting of distinct nodes v for each (red) objection link in the diagram.
4. E is the smallest set such that for all v, v' in G:
   (a) If $v \in I \cup RA \cup CA$ and $v' \in RA$ then:
      i. $(v, v') \in E$ if the link or box corresponding to v is at the beginning (bottom) of the reason link corresponding to v';

ii. $(v', v) \in E$ if the link or box corresponding to v is at the end (top) of the reason link corresponding to v';

(b) If $v \in I \cup RA \cup CA$ and $v' \in CA$ then:

i. $(v, v') \in E$ if the link or box corresponding to v is at the beginning (bottom) of the opposition link corresponding to v';

ii. $(v', v) \in E$ if the link or box corresponding to v is at the end (top) of the opposition link corresponding to v';

In practice, this translation involves uploading a Rationale file to the AIFdb, which automatically adds the Rationale arguments to the database. Figure 10 shows the argument from Figure 9 as rendered by the AIFdb viewer after upload (the argument can also be viewed at `http://www.arg.dundee.ac.uk/AIFdb/argview/737`).

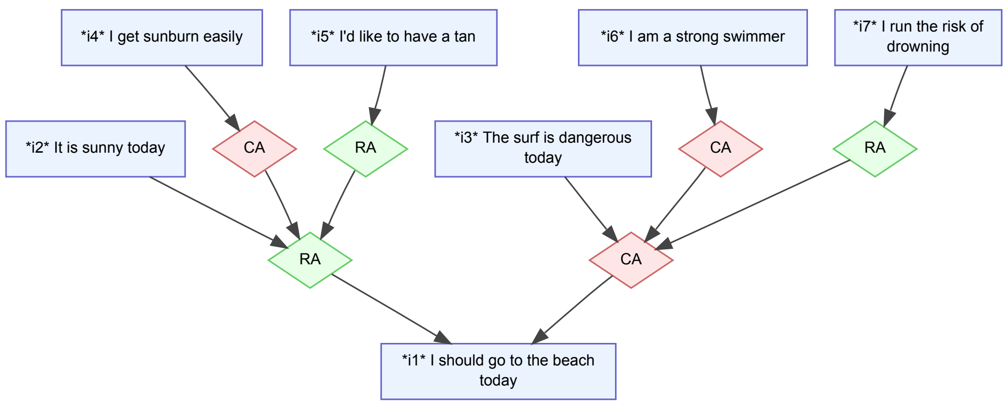

Figure 10: The beach argument in AIF

Now, if we take the statements $i1$ – $i3$ from the diagram in Figure 10, through subsequently applying Definitions 6.1 and 4.1 we get the following ASPIC$^+$ argumentation theory.

- $K_p = \{i_2, i_3\}$;
- $\mathcal{R}_d = \{r_1 : i_2 \Rightarrow i_1\}$;
- $\overline{i_1} = \{i_3\}$;
- arguments:
  $A_1 : i_2$, $A_2 : A_1 \Rightarrow_{r1} i_1$
  $B_1 : i_3$

Here, B_1 rebut-attacks A_2. This interpretation assumes that the user is implicitly expressing that the very fact that 'The surf is dangerous today' invalidates the conclusion that 'I should go to the beach today'. Such an assumption reflects argumentation as it often occurs in human practice, in the sense that the rationale for why a given conclusion invalidates another is not explicitly articulated: in this case that "since the surf is dangerous today then I *should not* go to the beach today and this takes precedence over the conclusion that 'I *should* go to the beach today' ". This rationale is encoded by specifying that i_3 ('The surf is dangerous today') is a contrary of i_1 ('I should go to the beach today'), and not vice versa. Hence, B_1's attack on A_2 is a (asymmetric) contrary-rebut, and so succeeds as a defeat independently of any preferences.

Rationale also allows the visualisation of reasons for, and objections to, links. Thus, one can visualise support for support links (akin to warrants [40]), such as the link from i_5 in Figure 10, support for objection links such as the link from i_7 in Figure 10, objections to support links (akin to undercutters [26]), such as the link from i_4 in Figure 9, and objections to objections (akin to attacks on attacks [20]), such as the link from i_6 in Figure 9.

Using Definitions 6.1 and 4.1, the full Figure 9 would be translated into ASPIC$^+$ as follows.

- $K_p = \{i_2, i_3, i_4, i_5, i_6, i_7\}$;
- $\mathcal{R}_d = \{r_1 : i_2 \Rightarrow i_1, r_2 : i_5 \Rightarrow r_1\}$;
- $\overline{i_1} = \{i_3\}, \overline{r_1} = \{i_4\}$;
- arguments:
  $A_1 : i_2$, $A_2 : A_1 \Rightarrow_{r1} i_1$
  $B_1 : i_3$
  $C_1 : i_4$
  $D_1 : i_5$, $D_2 : i_5 \Rightarrow_{r2} r_1$
  $E_1 : i_6$
  $F_1 : i_7$

Again, B_1 contrary-rebuts A_2. However, in this case C_1 also undercut-attacks A_2 and rebut-attacks D_2. Note that not all the information in the Figure 9 has been translated to ASPIC$^+$. In the intermediate AIF graph, the supporting effect of i_7 on the objection link from i_3 to i_1 is translated as a reason (through an RA-node) for the CA-node representing this objection link. However, this cannot be translated into ASPIC$^+$ directly, as the framework does not allow for reasons for contrary relations. The same is the case for the objection i_6 to the objection link from i_3 to i_1. In the AIF graph, this is represented as a conflicting element i_6 that is in conflict with (through a CA-node) the conflicted element that is the CA-node representing the objection link. However, ASPIC$^+$ does not allow us to express propositions that are contrary to the contrariness relations themselves.

If we want to fully translate the Rationale diagram, we therefore need to interpret it differently. For example, we could argue that 'The surf is dangerous today' (i_3) is a reason for i_{1neg}: 'I *should not* go to the beach today' through some RA-node: $i_3 \longrightarrow RA_3 \longrightarrow i_{1neg}$. Node i_7 then supports RA_3 through an RA-node of its own: $i_7 \longrightarrow RA_4 \longrightarrow RA_3$. This gives us two ASPIC$^+$ arguments: $B_1' : i_3 \Rightarrow_{r3} i_{1neg}$ (which attacks A_2) and $F_1' : i_7 \Rightarrow_{r4} r_3$, where r_3 and r_4 are based on RA_3 and RA_4, respectively. Thus, reasons that have an indirect supporting effect on the decision to *not* go to the beach are formulated.[10] Conversely, in the case of i_6 we can say that this somehow indirectly supports or expresses a preference for going to the beach. If we want to interpret this "objection against an objection" as a preference, we need to look at the way in which E-ASPIC$^+$ ultimately handles preference as attacks on attacks (Definition 3.12). 'I am a strong swimmer' (i_6) is then the basis of some argument that makes us prefer A_2 over B_1' or, in other words, i_6 is a reason for the preference $i_1 > i_{1neg}$.

This discussion and consideration of the translation options illustrates that if we want to use the current translation functions from AIF to ASPIC$^+$, in some cases we need to establish the interpretations that the original authors of the arguments had in mind when diagramming the arguments. The various options for translation also suggest the possibility of extending these functions: nodes that are in conflict with a conflict itself may point to preferences and nodes that support conflict may be interpreted as supporting reasons. However, it might also be decided that in this respect (E-)ASPIC$^+$ needs to be extended to further incorporate the various ways in which people naturally reason in an informal setting provided by a diagramming tool.

7 Conclusions and future research

In this paper we have shown how argument graphs as defined by the AIF can be understood as structures of the (E-)ASPIC$^+$ framework. We have demonstrated how a logical framework

[10]Note, however, that F_1' has no direct effect on the dialectical status of B_1'.

can aid in tracing possible inconsistencies in an AIF graph. Because of previous work showing relationships between (E-)ASPIC$^+$ and other logical formalisms, we have also implicitly shown the connection between the AIF and these other systems. In addition to the (E-)ASPIC$^+$ framework's obvious relation to [14, 26, 44, 28], several other well-known argumentation systems (e.g. [11],[6]) are shown by [27, 23] to be special cases of the (E-)ASPIC$^+$ framework. The connection between the AIF and (E-)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 argumentation formalisms that fall outside the scope of (E-)ASPIC$^+$ (see [27, 29] for examples); this would also further clarify the relation between the (E-)ASPIC$^+$ framework and these other formalisms.

In section 6 we lay the foundations for evaluating arguments in diagramming tools according to argument acceptability semantics. This thus shows a possible use of the AIF as a bridge between natural argumentation as performed by humans, and logical models of argumentation. This in turn provides foundations for logic-based normative concepts of reasoning to guide human argumentation (e.g. by suggesting an argument the user needs to attack in order to reinstate his main point), as well as for integrating human argumentation with logic based models of reasoning.

A valid question here is whether we really need an intermediate language like the AIF ontology to stand between our models of natural argument and our logical models of argument. ASPIC$^+$ itself is based on models from philosophy [26] and legal reasoning [28], models which have been expressly developed to understand and specify natural human reasoning. Argumentation schemes can be represented in the ASPIC$^+$ framework through the use of rule schemes, and its tree-structured arguments are very similar to, for example, Araucaria-style argument diagrams [9]. However, the heritage of ASPIC$^+$ lies in systems of logic. As a result, it inherits a specific focus and approach. Semantics, for example, are those of propositional logic, in which every formula means one of two things, true or false. Its logical engine is designed with a single purpose in mind: determination of entailment (and by extension, some sense of consistency). And it assumes a propositional foundation in which deixis is resolved, repetition prohibited and rephrasing irrelevant. The AIF ontology's focus is on real text, in which quotation is common, straw man arguments are subtle and effective and repetition is a rhetorically powerful technique.

The paper shows that a relatively simple AIF argument graph contains enough information for representation in a mature logical framework such as ASPIC$^+$. Information that is important for logical systems but that is not contained in the graph, such as defeat relations for example, can be calculated from information that is in the graph. 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 a calculated 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.[11]

A connection between computational argumentation theory and argumentation practice is vital if the former is to find application and real world utility in the latter. Of course, there are other areas where computational argument might have impact (in distributed, complex systems; in automated markets; etc.) but with so much computational work – from [16] to [14] – explicitly inspired by and acknowledging influence from natural argumentation, it is an enormous missed opportunity if we, as a community, fail to connect computational with natural practice. Yet to date, there has been only the lightest connections between formal and natural models. Specifically, only a very small subset of features of argument (typically those concerned with logical relations between propositions) have found their way into computational systems. In this paper we have shown for the first time how not just a single technique, but an wide range

[11] An implementation of this connection between an extended version of AIF and abstract argumentation has been trialled in OVA-gen, a tool for computing acceptability semantics accessible at *http://www.arg.dundee.ac.uk/OVA/* [www.arg.dundee.ac.uk].

of tools, theories and systems encompassed by or compatible with the AIF ontology (including, but not limited to, diagramming systems such as Rationale [5], philosophical theories of argumentation such as [26], linguistic theories that apply to argument such as [37] and recent advances in argumentation theory itself such as [45]) can be connected to work not only on structured argumentation, characterised by ASPIC$^+$, but also on abstract argumentation via the formal machinery introduced in [27]. Thus we have laid a foundation for now exploring what our formal and computational models can do in natural argument contexts, and, similarly, exploring what features of natural argument might next be tackled to enrich our computational systems. In this way we aim to contribute both to the continuing growth of research in computational models of argument whilst simultaneously contributing to the relevance and applicability of that research.

Appendix: Proofs

In order to be able to prove identity-preserving translations for ASPIC$^+$, we use the following conditions and assumptions on the graph G.

- Definition 2.1(3): Only I-nodes are initial nodes.
- Definition 2.1(4): RA-nodes are connected to at least one direct predecessor via a *premise* edge and to exactly one direct successor via a *conclusion* edge.
- Definition 2.1(5): PA-nodes are connected to exactly one direct predecessor via a *preferred element* edge and to exactly one direct successor via a *dispreferred element* edge.
- Definition 2.1(6): CA-nodes are connected to exactly one direct predecessor via a *conflicting element* edge and to exactly one direct successor via a *conflicted element* edge.
- A1: (Assumption 4.3) There are no PA- or CA-nodes connected to an RA-node via a *premise* or *conclusion* edge.
- A2: (Assumption 4.3) There are no PA- or CA-nodes connected to a PA-node via a *preferred element* or *dispreferred element* edge.
- A3: (Assumption 4.3) There are no PA- or CA-nodes connected to a CA-node via a *conflicting element* or *conflicted element* edge.

Under these assumptions it can be proven that all translations from the AIF to ASPIC$^+$ and then back result in an AIF graph that is isomorphic with the original graph in that the graphs differ at most in their names for the nodes.

We first prove that under the above assumptions any node or edge in an AIF graph G is translated to something in the AT based on G and any element of a component in the corresponding ASPIC$^+$ AT is the result of a translation from G.

Lemma 7.1 If AT is an ASPIC$^+$ argumentation theory based on AIF graph G, then:

1. $i \in I$ iff $i \in \mathcal{L} \setminus \mathcal{L}_R$.
2. $r \in RA$ iff $r \in \mathcal{L}_R$.
3. For any $i \in I$ it holds that $i \in \mathcal{K}$ or i is an antecedent or the consequent of a rule in $\mathcal{R}$.
4. For any $i \in \mathcal{K}$ it holds that $i \in I$.
5. For any $r \in RA$ of form *strict/defeasible* and any $v \in I \cup RA$, if $(v, r) \in E$, then there exists a unique inference rule $r : v_1, \ldots, v_m \rightarrow/\Rightarrow v_n \in \mathcal{R}$ such that $v = v_1$ or $\ldots$ or $v = v_m$.

6. For any $r \in RA$ of form *strict/defeasible* and any $v \in I \cup RA$, if $(r, v) \in E$, then there exists a unique inference rule $r : v_1, \ldots, v_n \rightarrow/\Rightarrow v \in \mathcal{R}$.

7. For any inference rule $r : v_1, \ldots, v_m \rightarrow/\Rightarrow v_n \in \mathcal{R}$ it holds that $v_i \in I$ if $v_i \in \mathcal{L}_I$ and $v_i \in RA$ if $v_i \in \mathcal{L}_R$, $r \in RA$ of form *strict/defeasible* and $(v_1, r), \ldots, (v_m, r), (r, v_n) \in E$.

8. For all $v \in V$ and $c \in CA$: if $(v, c) \in E$ then for some v' it holds that $(v, v') \in {}^-$.

9. For all $v \in V$ and $c \in CA$: if $(c, v) \in E$ then for some v' it holds that $(v', v) \in {}^-$.

10. For all $c \in CA$ there exist unique $v, v' \in I \cup RA$ such that $(v, c) \in E$ and $(c, v') \in E$ and $(v, v') \in {}^-$.

11. For all $(v, v') \in {}^-$ it holds that $v, v' \in I \cup RA$.

12. For all $(v, v') \in {}^-$ there exists a unique $c \in CA$ such that $(v, c) \in E$ and $(c, v') \in E$.

13. For all $i \in I$ and $p \in PA$: if $(i, p) \in E$ then for some $i' \in I$ it holds that $(i, i') \in \leq'$.

14. For all $i \in I$ and $p \in PA$: if $(p, i) \in E$ then for some $i' \in I$ it holds that $(i', i) \in \leq'$.

15. For all $r \in RA$ and $p \in PA$: if $(r, p) \in E$ then for some $r' \in RA$ it holds that $(r, r') \in \leq$.

16. For all $r \in RA$ and $p \in PA$: if $(p, r) \in E$ then for some $r' \in RA$ it holds that $(r', r) \in \leq$.

17. For all $p \in PA$ there exist unique $v, v' \in I \cup RA$ such that $(v, p) \in E$ and $(p, v') \in E$ and $(v', v) \in \leq'$ or $(v', v) \in \leq$.

18. For all $(v, v') \in \leq'$ it holds that $v, v' \in I$.

19. For all $(v, v') \in \leq$ it holds that $v, v' \in RA$.

20. For all $(v, v') \in \leq'$ there exists a unique $p \in PA$ such that and $(v, p) \in E$ and $(p, v') \in E$.

21. For all $(v, v') \in \leq$ there exists a unique $p \in PA$ such that $(v, p) \in E$ and $(p, v') \in E$.

Proof:

1. By the definition of $\mathcal{L}$ in Definition 4.1(1).

2. By the definition of $\mathcal{L}$ in Definition 4.1(1).

3. By construction of $\mathcal{K}$ and $\mathcal{R}$ (Definition 4.1(2,3)) and Definition 2.1(4).

4. By construction of $\mathcal{K}$ in Definition 4.1(2) and Definition 2.1(3).

5. By construction of $\mathcal{R}$ in Definition 4.1(3) and Definition 2.1(4).

6. By construction of $\mathcal{R}$ in Definition 4.1(3) and Definition 2.1(4).

7. By construction of $\mathcal{R}$ in Definition 4.1(3) and assumption A1.

8. By construction of ${}^-$ in Definition 4.1(4).

9. By construction of ${}^-$ in Definition 4.1(4).

10. From (8,9), assumptions A1, A2 and A3 and Definition 2.1(6).

11. From Definition 4.1(1,4) and assumption A3.

12. From the construction of $^-$ in Definition 4.1(4) and Definition 2.1(6).

13. By construction of $\leq'$ in Definition 4.1(5) and assumptions A1 and A2.

14. By construction of $\leq'$ in Definition 4.1(5) and assumptions A1 and A2.

15. By construction of $\leq$ in Definition 4.1(6) and assumption A1.

16. By construction of $\leq$ in Definition 4.1(6) and assumption A1.

17. From 13-16, assumptions A1, A2 and A3 and Definition 2.1(5).

18. From Definition 4.1(1,2,5), assumption A2 and Definition 2.1(4, 5).

19. From Definition 4.1(1,6), assumption A2.

20. From the construction of $\leq'$ in Definition 4.1(5) and Definition 2.1(5) and assumptions A1 and A2.

21. From the construction of $\leq$ in Definition 4.1(6) and Definition 2.1(5) and assumptions A1 and A2.

Next, consider for any ASPIC$^+$ argumentation theory AT based on an AIF graph G the set $Args_{AT}$, that is, the set of all arguments that can be constructed on the basis of AT. When in Definition 4.2 we choose $\mathcal{A}_{AT} = Args_{AT}$, it can be proven that if AT is based on G, then Definition 4.2 returns an AIF graph that differs at most from G in the names for the nodes and edges of G.

Theorem 4.4. *Let G' be an AIF graph, and AT be the ASPIC$^+$ argumentation theory based on G'. Let G be an AIF graph based on $Args_{AT}$. Then G is isomorphic to G'.*

Proof: We first prove that G is an AIF graph based on $Args_{AT}$. Then the result follows from the observation that any other G' based on $Args_{AT}$ differs at most from G by uniformly substituting names of nodes in G'.

Let $G = (V, E)$ where $V = I \cup RA \cup CA \cup PA$. Note that all these elements of G are defined in Definition 4.2. Let $G' = (V', E')$ where $V' = I' \cup RA' \cup CA' \cup PA'$. We prove that $I' = I$, $RA' = RA$, $CA' = CA$, $PA' = PA$ and $E' = E$. We consider all cases of Definition 4.2 in turn.

1. Note first that $\mathcal{A} = Args_{AT}$ and by construction of AT all rules in $\mathcal{R}$ are used in at least one argument, so $\texttt{Rules}(\mathcal{A}) = \mathcal{R}$. Then $\texttt{Wff}(\mathcal{A})$ consists of $\mathcal{K}$ plus all antecedents and consequents of any rule in $\mathcal{R}$. Moreover, by Lemma 7.1(4,7) all elements of $\mathcal{K}$ and antecedents and consequents of any rule in $\mathcal{R}$ are in $I' \cup RA'$. Therefore $I \subseteq I'$.

   Next, by Lemma 7.1(3) any $i \in I'$ is in $\mathcal{K}$ or is an antecedent of consequent of a rule in $\mathcal{R}$. Then since each element of $\mathcal{K}$ is in $Args_{AT}$ and all rules in $\mathcal{R}$ are used in at least one argument in $Args_{AT}$, we have that $I' \subseteq I$. But then $I = I'$.

2. By Lemma 7.1(5,6) for any $r \in RA'$ there exists a unique rule named r in $\mathcal{R}$, so $RA' \subseteq RA$. Next, by Lemma 7.1(7) for any rule named r in $\mathcal{R}$ it holds that $r \in RA'$, so $RA \subseteq RA'$. But then $RA' = RA$.

3. By Lemma 7.1(10) for all $c \in CA'$ there exist a unique pair $(\varphi, \psi) \in {}^-$ such that (φ, c) and (c, ψ) are in E'. Moreover, by Lemma 7.1(12) for any pair $(\varphi, \psi) \in {}^-$ there exists a unique $c' \in CA'$ such that (φ, c') and (c', ψ) are in E'. Then $c = c'$. Then choosing $v = c$ for all such pairs in Definition 4.2(3) yields $CA' = CA$.

4. By Lemma 7.1(17) for all $p \in PA'$ there exist a unique pair (φ, ψ) in either $\leq'$ or $\leq$. Moreover, by Lemma 7.1(20,21) for any pair (φ, ψ) in $\leq'$ or $\leq$ there exists a unique $p' \in PA'$ such that (φ, p') and (p', ψ) are in E'. Then $p = p'$. Then choosing $v = p$ for all such pairs in Definition 4.2(4) yields $PA' = PA$.

   For cases (5a-5c) the following notation is used: for any set of edges E and set of nodes N, $E \mid_N = \{(v, v') \in E \mid v \in N \text{ or } v' \in N\}$.

5. (a) Let $v \in I' \cup RA'$ and $r \in RA'$. By Lemma 7.1(5) all relations $(v, r) \in E'$ are such that v is an antecedent of $r \in \mathcal{R}$. Likewise, by Lemma 7.1(6) all relations $(r, v) \in E'$ are such that v is the consequent of $r \in \mathcal{R}$. Then $(v, r) \in E$ and $(r, v) \in E$, so with (1) we have $E' \mid_{I'} \subseteq E \mid_I$. Moreover, if v is an antecedent of $r \in \mathcal{R}$ then by Lemma 7.1(7), $(v, r) \in E'$. Likewise, if v is the consequent of $r \in \mathcal{R}$ then by Lemma 7.1(7), $(r, v) \in E'$, so with (1) we have $E \mid_I \subseteq E' \mid_{I'}$. But then $E' \mid_{I'} = E \mid_I$.

   (b) Let $c \in CA'$. By Lemma 7.1(10) all pairs of pairs $(v, c), (c, v')$ in E' are such that there exists a unique pair $(v, v') \in {}^-$. Moreover, by Lemma 7.1(12) for every pair $(v, v') \in {}^-$ there exists a unique $c' \in CA'$ such that $(v, c') \in E'$ and $(c', v') \in E'$. Then $c = c'$. Note that under (3) of this proof v in Definition 4.2(3) was chosen to be c. Then with (3) we have $E' \mid_{CA'} = E \mid_{CA}$.

   (c) Let $p \in PA'$. By Lemma 7.1(13-16) all pairs of pairs $(v, p), (p, v')$ in E' are such that there exists a unique pair (v, v') in either $\leq$ or $\leq'$. Moreover, by Lemma 7.1(20,21) for every pair (v, v') in $\leq$ or $\leq'$ there exists a unique $p' \in PA'$ such that $(v, p') \in E'$ and $(p', v') \in E'$. Then $p = p'$. Note that under (4) of this proof v in Definition 4.2(3) was chosen to be p. Then with (4) we have $E' \mid_{PA'} = E \mid_{PA}$.

   From (5a-c) it follows that $E' = E$. Moreover, it is straightforward to prove that all edges have the same type in E as in E'.

In order to be able to prove identity-preserving translations for E-ASPIC$^+$, we also use the conditions (3 – 6) from Defintion 2.1. However, the assumptions A1 – A3 are adjusted.

- A1$'$: (Assumption 4.3) There are no CA-nodes connected to an RA-node via a *premise* or *conclusion* edge.
- A2$'$: (Assumption 4.3) There are no CA-nodes connected to a PA-node via a *preferred element* or *dispreferred element* edge.
- A3$'$: (Assumption 4.3) There are no CA-nodes connected to a CA-node via a *conflicting element* or *conflicted element* edge.

Lemma 7.1 can now be simplified in that E-ASPIC$^+$ has no input orderings but it must be adjusted to account for the possibility that facts and rules can be or contain preference expressions.

Lemma 7.2 If AT is an E-ASPIC$^+$ argumentation theory based on AIF graph G, then:

1. $i \in I$ iff $i \in \mathcal{L} \setminus \mathcal{L}_R \setminus \mathcal{L}_m$.
2. $r \in RA$ iff $r \in \mathcal{L}_R$.
3. For any $i \in I$ it holds that $i \in \mathcal{K}$ or i is an antecedent or the consequent of a rule in $\mathcal{R}$.
4. For any $v \in \mathcal{K}$ it holds that $v \in I$ or v is based on a $p \in PA$.

5. For any $r \in RA$ of form *strict/defeasible* and any $v \in I \cup RA \cup PA$, if $(v, r) \in E$ and is of type *premise*, then there exists a unique inference rule $r : v_1, \ldots, v_m \rightarrow/\Rightarrow v_n \in \mathcal{R}$ such that $v = v_1$ or ... or $v = v_m$ or v_1 is based on v or ... or v_m is based on v.

6. For any $r \in RA$ of form *strict/defeasible* and any $v \in I \cup RA \cup PA$, if $(r, v) \in E$ and is of type *conclusion*, then there exists a unique inference rule $r : v_1, \ldots, v_m \rightarrow/\Rightarrow v_n \in \mathcal{R}$ such that $v_n = v$ or v_n is based on v.

7. For any inference rule $r : v_1, \ldots, v_m \rightarrow/\Rightarrow v_n \in \mathcal{R}$ it holds that $v_i \in I$ if $v_i \in \mathcal{L}_I$, $v_i \in RA$ if $v_i \in \mathcal{L}_R$, and $p \in PA$ if $v_i \in \mathcal{L}_m$ and is based on p, and $r \in RA$ of form *strict/defeasible* and $(v_1^*, r), \ldots, (v_m^*, r), (r, v_n^*) \in E$, where $v_i^* = v_i$ or v_i^* is based on v_i.

8. For all $v \in V$ and $c \in CA$: if $(v, c) \in E$ then for some v' and v'' it holds that $(v'', v') \in {}^-$ where $v'' = v$ or v'' is based on v.

9. For all $v \in V$ and $c \in CA$: if $(c, v) \in E$ then for some v' and v'' it holds that $(v', v'') \in {}^-$ where $v'' = v$ or v'' is based on v.

10. For all $c \in CA$ there exist unique $v, v' \in I \cup RA \cup PA$ such that $(v, c) \in E$ and $(c, v') \in E$ and $(w, w') \in {}^-$ such that $w/w' = v/v'$ or w/w' is based on v/v'.

11. For all $(v, v') \in {}^-$ it holds that $v/v' \in I \cup RA$ or v/v' is based on w/w' and $w/w' \in PA$.

12. For all $(v, v') \in {}^-$ there exists a unique $c \in CA$ such that $(v, c)/(c, v') \in E$ or c/v' is based on w/w' and $(w, c)/(c, w') \in E$.

13. $\mathcal{L}_m \cap \mathtt{Wff}(\mathcal{A}) = \mathcal{L}_m$.

14. For all $v \in V$ and $p \in PA$: if $(v, p) \in E$ and is of type *preferred*, then for some v' and v'' it holds that $v' > v'' \in \mathcal{L}_m$ is based on p and $v' = v$ or v' is based on v.

15. For all $v \in V$ and $p \in PA$: if $(p, v) \in E$ and is of type *dispreferred*, then for some v' and v'' it holds that $v'' > v' \in \mathcal{L}_m$ is based on p and $v' = v$ or v' is based on v.

16. For all $p \in PA$ there either exist unique $v, v' \in I \cup RA$ such that $(v, p) \in E$ and $(p, v') \in E$ and $v > v' \in \mathcal{L}_m^0$ and $v > v'$ is based on p; or there exist unique $p', p'' \in PA$ such that $(p', p) \in E$ and $(p, p'') \in E$ and there exists a unique $\varphi > \psi \in \mathcal{L}_m$ based on p such that φ is based on p' and ψ is based on p''.

17. For all $v > v' \in \mathcal{L}_m$ there exists a unique $p \in PA$ such that $v > v'$ is based on p and $(v^*, p) \in E$ and $(p, v'^*) \in E$ where $v^*/v'^* = v/v'$ if $v/v' \in \mathcal{L}_o$ or $v^*/v'^* = p'/p''$ if $v/v' \in \mathcal{L}_m$ and v/v' is based on $p'/p'' \in PA$.

Proof:

1. By the definition of $\mathcal{L}$ in Definition 5.1(1).
2. By the definition of $\mathcal{L}$ in Definition 5.1(1).
3. By construction of $\mathcal{K}$ and $\mathcal{R}$ in Definition 5.1(2,3) and Definition 2.1(4).
4. By construction of $\mathcal{K}$ in Definition 5.1(2) and Definition 2.1(3, 4).
5. By construction of $\mathcal{R}$ in Definition 5.1(3) and Definition 2.1(4).
6. By construction of $\mathcal{R}$ in Definition 5.1(3) and Definition 2.1(4).
7. By construction of $\mathcal{R}$ in Definition 5.1(3) and assumption A1$'$.

8. By construction of $^{-}$ in Definition 5.1(4).

9. By construction of $^{-}$ in Definition 5.1(4).

10. From (8,9), assumptions A1$'$, A2$'$ and A3$'$ and Definition 2.1(6).

11. From Definition 5.1(1,4) and assumption A3$'$.

12. From the construction of $^{-}$ in Definition 5.1(4) and Definition 2.1(6).

13. Since $\mathcal{A} = Args_{AT}$ and by construction of AT all rules in $\mathcal{R}$ are used in at least one argument, we have that $\texttt{Rules}(\mathcal{A}) = \mathcal{R}$. Then $\texttt{Wff}(\mathcal{A})$ consists of $\mathcal{K}$ plus all antecedents and consequents of any rule in $\mathcal{R}$. We next prove that any element φ of $\mathcal{L}_m$ is in $\mathcal{K}$ or is an antecedent or a consequent of a rule in $\mathcal{R}$. By Lemma 7.2(17) we have that φ is based on a $p \in PA'$. Next, by construction of $\mathcal{K}$, for any such p that has no incoming RA node in G we have that $\varphi \in \mathcal{K}$ while by construction of $\mathcal{R}$ and condition (4) (Definition 2.1), for any such p that has a incoming RA node in G we have that φ is an antecedent or a consequent of a rule in $\mathcal{R}$.

14. By construction of $\mathcal{L}_m$ in Definition 5.1(1b).

15. By construction of $\mathcal{L}_m$ in Definition 5.1(1b).

16. From (14,15), the construction of $\mathcal{L}_m$ in Definition 5.1(1b) and Definition 2.1(5).

17. From the construction of $\mathcal{L}_m$ in Definition 5.1(1b) and Definition 2.1(5).

In proving the translation results the proof of Theorem 4.4, no preference relations in E-ASPIC$^+$ need to be considered but in turn it must now be proven that when the translation from E-ASPIC to AIF involves formulas from $\mathcal{L}_m$, the original PA-nodes and edges involving PA-nodes are returned.

Theorem 5.4. *Let G' be an AIF graph, and EAT be the E-ASPIC$^+$ argumentation theory based on G'. Let G be an AIF graph based on $Args_{EAT}$. Then G is isomorphic to G'.*

Proof: The proof of Theorem 4.4 must be adjusted as follows.

1. As for Theorem 4.4(1), replacing Lemmas 7.1(3), 7.1(4) and 7.1(7) by Lemmas 7.2(3), 7.2(4) and 7.2(7) and taking into account that elements of $\mathcal{K}$ and antecedents and consequents of any rule in $\mathcal{R}$ may correspond to a node in PA instead of themselves being in $I' \cup RA'$.

2. By Lemma 7.2(5,6) and Definition 2.1(4) it holds that for any $r \in RA'$ there exists a unique rule named r in $\mathcal{R}$, so $RA' \subseteq RA$. Next, by Lemma 7.2(7) for any rule named r in $\mathcal{R}$ it holds that $r \in RA'$, so $RA \subseteq RA'$. But then $RA' = RA$.

3. As for Theorem 4.4(3), replacing Lemmas 7.1(10) and 7.1(12) by Lemmas 7.2(10) and 7.2(12) and taking into account that φ and ψ may be based on nodes in CA' instead of themselves being in CA'.

4. Note first according to Lemma 7.2(13) we have $\mathcal{L}_m \cap \texttt{Wff}(\mathcal{A}) = \mathcal{L}_m$. Next, consider any $p \in PA'$. By Lemma 7.2(16) there exist a unique formula $\varphi > \psi \in \mathcal{L}_m$ based on p. By Lemma 7.2(17) for all $\varphi > \psi \in \mathcal{L}_m$ there exists a unique $p' \in PA'$ such that $\varphi > \psi$ is based on p'. So $p = p'$. Then choosing $v = p$ for all such pairs in Definition 5.1(4) yields $PA'_i = PA_i$.

5. (a) Let $v \in I' \cup RA'$ and $r \in RA'$. By Lemma 7.2(5) all relations $(v, r) \in E'$ of type *premise* are such that v is an antecedent of $r \in \mathcal{R}$. Likewise, by Lemma 7.2(6) all relations $(r, v) \in E'$ of type *conclusion* are such that v is the consequent of $r \in \mathcal{R}$. Then $(v, r) \in E$ and $(r, v) \in E$, so with (1) we have $E' \mid_{I'} \subseteq E \mid_I$. Moreover, if v is an antecedent of $r \in \mathcal{R}$ then by Lemma 7.2(7), $(v, r) \in E'$. Likewise, if v is the consequent of $r \in \mathcal{R}$ then by Lemma 7.2(7), $(r, v) \in E'$, so with (1) we have $E \mid_I \subseteq E' \mid_{I'}$. But then $E' \mid_{I'} = E \mid_I$.

   (b) As for Theorem 4.4(5b), replacing Lemmas 7.1(10) and 7.1(12) by Lemmas 7.2(10) and 7.2(12) and taking into account that v, v' (such that $(v, v') \in {}^-$) may be based on nodes in PA' instead of themselves being in $I' \cup RA'$.

   (c) Let $p \in PA'$. By Lemma 7.2(14,15) all pairs of pairs $(v, p), (p, v')$ in E' of type *preferred*, respectively, *dispreferred* are such that there exists a unique formula $w > w'$ in $\mathcal{L}_m$ based on p and $w/w' = v/v'$ or w/w' is based on v/v'. Moreover, by Lemma 7.2(17) for every formula formula $w > w'$ in $\mathcal{L}_m$ there exists a unique $p' \in PA$ such that $w > w'$ is based on p' and is such that $(w^*, p') \in E$ and $(p', w'^*) \in E$, where $w^*/w'^* = w/w'$ if $w/w' \in \mathcal{L}_o$ or $w^*/w'^* = q/q'$ if $w/w' \in \mathcal{L}_m$ and w/w' is based on $q/q' \in PA$. Then $p = p'$. Note that under (4) of this proof v in Definition 5.1(4) was chosen to be p. Then with (4) we have $E' \mid_{PA'} = E \mid_{PA}$.

   From (5a-c) it follows that $E' = E$. Moreover, it is straightforward to prove that all edges have the same type in E as in E'.

References

[1] L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney andS. 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] L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *International Journal of Automated Reasoning*, Volume 29 (2):125–169, 2002.

[3] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–216, 2002.

[4] T.J.M. Bench-Capon and P.E. Dunne (eds.). Special issue on Argumentation in Artificial Intelligence. *Artificial Intelligence*, 171, 2007.

[5] T. Berg, T. van Gelder, F. Patterson, and S. Teppema. *Critical Thinking: Reasoning and Communicating with Rationale*. Amsterdam: Pearson Education Benelux, 2009.

[6] Ph. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, Cambridge, MA, 2008.

[7] F.J. Bex, J. Lawrence, and C.A. Reed. The argument interchange format (aif) specification. Technical report, University of Dundee, 2011.

[8] F.J. Bex, H. Prakken, and C.A. Reed. A formal analysis of the AIF in terms of the ASPIC framework. In P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari, editors, *Computational Models of Argument. Proceedings of COMMA 2010*, pages 99–110, Amsterdam, The Netherlands, 2010. IOS Press.

[9] F.J. Bex, H. Prakken, C.A. Reed, and D.N. Walton. Towards a Formal Account of Reasoning about Evidence: Argumentation Schemes and Generalisations. *Artificial Intelligence and Law*, 11(2/3):125–165, 2003.

[10] F.J. Bex and C.A. Reed. Schemes of Inference, Conflict and Preference in a Computational Model of Argument. *Studies in Logic, Grammar and Rhetoric*, 36, 2011.

[11] 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.

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

[13] 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.

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

[15] A.J. Garcia and G.R. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4:95–138, 2004.

[16] T.F. Gordon, H. Prakken, and D.N. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896, 2007.

[17] T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43:907–928, 1995.

[18] J. Lawrence, F. Bex, M. Snaith, and C. Reed. Aifdb: Infrastructure for the argument web. In *Computational Models of Argument. Proceedings of COMMA 2012*, 2012. to appear.

[19] J. Lawrence, F.J. Bex, and C. Reed. Dialogues on the argument web: Mixed initiative argumentation with arvina. In *Computational Models of Argument. Proceedings of COMMA 2012*, 2012. to appear.

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

[21] S. Modgil and J. McGinnis. Towards characterising argumentation based dialogue in the argument interchange format. In *4th International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2007)*, pages 80–93, 2007.

[22] S. Modgil and H. Prakken. Reasoning about preferences in structured extended argumentation frameworks. In P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari, editors, *Computational Models of Argument. Proceedings of COMMA 2010*, pages 347–358, Amsterdam, The Netherlands, 2010. IOS Press.

[23] S. Modgil and H. Prakken. Revisiting preferences and argumentation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1021–1026, 2011.

[24] S. Modgil and H. Prakken. A general account of argumentation and preferences. Technical report, www.dcs.kcl.ac.uk/staff/smodgil/GAP.pdf, 2012.

[25] R. Mochales Palau and M.F. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19:1–20, 2011.

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

[27] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.

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

[29] H. Prakken and G.A.W. Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 219–318. Kluwer Academic Publishers, Dordrecht/Boston/London, second edition, 2002.

[30] 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.

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

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

[33] C.A. Reed and D.P. Long. Content ordering in the generation of persuasive discourse. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI97)*, pages 1022–1027, 1997.

[34] C.A. Reed and G. Rowe. Araucaria: Software for Argument Analysis, Diagramming and Representation. *International Journal of AI Tools*, 13(4):961–980, 2004.

[35] C.A. Reed, S. Wells, K. Budzynska, and J. Devereux. Building arguments with argumentation : the role of illocutionary force in computational models of argument. In *Proceedings of COMMA 2010*, 2010.

[36] R Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[37] J. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[38] M. Snaith, J. Lawrence, and C. Reed. Pipelining argumentation technologies. In P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari, editors, *Computational Models of Argument. Proceedings of COMMA 2010*, pages 447–454, Amsterdam, The Netherlands, 2010. IOS Press.

[39] P. St. Dizier. Processing natural language arguments with the <textcoop> platform. *Argument and Computation*, 3:49–83, 2012.

[40] S.E. Toulmin. *The Uses of Argument*. Cammbridge University Press, Cambridge, 1958.

[41] B. van Gijzel and H. Prakken. Relating Carneades with abstract argumentation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1013–1019, 2011.

[42] B. Verheij. Deflog: on the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation*, 13:319–346, 2003.

[43] J. Visser, F.J. Bex, C.A. Reed, and B. Garssen. Correspondence between the pragma-dialectical discussion model and the argument interchange format. *Studies in Logic, Grammar and Rhetoric*, 36, 2011.

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

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