On support relations in abstract argumentation as abstractions of inferential relations

Henry Prakken

Abstract. Arguably the significance of an abstract model of argumentation depends on the range of realistic instantiations it allows. This paper therefore investigates for three frameworks for abstract argumentation with support relations whether they can be instantiated with the ASPIC+ framework for structured argumentation. Both evidential argumentation systems and a simple extension of Dung’s abstract frameworks with support relations proposed by Dung & Thang (2014) are shown to allow such an instantiation. However, for bipolar argumentation frameworks a positive result is only obtained for variants with only direct and secondary attacks; counterexamples are provided for variants with supported attacks, even for the special case of deductive support.

1 Introduction

There have been several recent proposals to extend [8]’s well-known abstract argumentation frameworks (AFs) with support relations. Among the best known are [5, 6]’s Bipolar Argumentation Frameworks (BAFs) and [19]’s Evidential Argumentation Systems (EASs). Arguably the significance of any abstract model of argumentation depends on the range of realistic instantiations it allows. Dung’s AFs score very well in this respect, since in [8] various systems of nonmonotonic logic, argumentation and logic programming are reconstructed as AFs, namely, [24]’s default logic, [19]’s argumentation system and two semantics for logic programming. Moreover, much later work on structured argumentation was formulated to generate AFs. For example, both [23]’s system and the ASPIC(+) framework [4, 21, 15, 16] were explicitly defined to generate Dung-style AFs, while assumption-based argumentation (ABA) as defined in [3] was in [9] proven to instantiate AFs.

All these instantiations of AFs define relations of inferential support between (sets of) formulas in terms of definitions of structured arguments. This raises the question whether BAFs and EASs can be seen as abstractions of the inferential support relations modelled in these approaches. The present paper aims to answer this question. It should be noted that the papers on BAFs and EASs do not address these approaches. The present paper aims to answer this question.

This question will be answered by investigating whether the ASPIC+ framework of [21, 15] can be reformulated as an instantiation of BAFs or EASs. The choice of ASPIC+ for these purposes is justified by its generality and the fact that it captures various other approaches as special cases, such as ABA as studied in [9], various forms of classical argumentation as studied in [11], and various instantiations with Tarskian abstract logics as studied by [1]. Therefore, results in terms of ASPIC+ are representative for a large class of argumentation systems.

Below first an alternative way of adding support relations to AFs called SuppAFs will be proposed, based on an idea of [10]. Then ASPIC+ will be shown to instantiate SuppAFs, after which a variant of BAFs with only direct and secondary attacks will turn out to be equivalent to SuppAFs and so also suitable as an abstraction of ASPIC+. The same will be shown for an abstract version of [20]’s recursive argument labellings. However, versions of BAFs with so-called supported attacks will be shown to be inadequate as abstractions of ASPIC+, even for the special case of deductive argumentation. Finally, ASPIC+ will (for preferred semantics) be shown to be translatable as a special case of EASs. The question then arises of what EASs add to SuppAFs as proposed in the present paper.

2 Formal preliminaries

We first review the formal frameworks investigated in this paper.

2.1 Abstract argumentation frameworks

An abstract argumentation framework (AF) is a pair \((A, D)\), where \(A\) is a set of arguments and \(D \subseteq A \times A\) is a binary relation of defeat.\(^1\) We say that \(A\) strictly defeats \(B\) if \(A\) defeats \(B\) while \(B\) does not defeat \(A\). A semantics for AFs returns sets of arguments called extensions, which are internally coherent and defend themselves against attack.

Definition 1 Let \((A, D)\) be an AF. For any \(X \in A\), \(X\) is acceptable w.r.t. some \(S \subseteq A\) iff \(\forall Y\ s.t. (Y, X) \in D\ implies \exists Z \in S\ s.t. (Z, Y) \in D\). Let \(S \subseteq A\) be conflict free, i.e., there are no \(A, B\) in \(S\) such that \((A, B) \in D\). Then \(S\) is: an admissible set iff \(X \in S\ implies \ X\ is acceptable\); a complete extension iff \(X \in S\ whenever \ X\ is acceptable\); a preferred extension iff it is a set inclusion maximal complete extension; the grounded extension iff it is the set inclusion minimal complete extension; a stable extension iff it is a preferred extension and \(\forall Y \notin S\ implies \exists X \in S\ s.t. (X, Y) \in D\).

For \(T \in \{\text{complete, preferred, grounded, stable}\}\), \(X\) is sceptically or credulously justified under the \(T\) semantics if \(X\) belongs to all, respectively at least one, \(T\) extension.

\(^1\) [8] calls defeat “attack” but in this paper “defeat” is used to be compatible with the terminology in ASPIC+.

\(^2\) Department of Information and Computing Sciences, Utrecht University and Faculty of Law, University of Groningen, The Netherlands, email: H.Prakken@uu.nl
2.2 Bipolar argumentation frameworks

Bipolar frameworks add a binary support relation $S$ to AFs. Thus BAfs are a triple $(A, D, S)$. In [5] but not in [6] $D$ and $S$ are assumed to be disjoint. In [5] a sequence of supports for argument $B$ by argument $A$ is a sequence $ASD_1, \ldots, ASD_nBD$ (it is said that $A$ supports $B$). A supported attack for argument $B$ by argument $A$ is a sequence $(A, X, B)$ of arguments such that $A$ supports $X$ and $X$ attacks $B$. A set $S \subseteq A$ is said to set-attack an argument $A \in A$ iff there exists a supported or direct attack on $A$ from an element of $S$. Finally, a set $S \subseteq A$ is a conflict-free iff there are no $A$ and $B$ in $S$ such that $\{A\}$ set-attacks $B$. While this BAfs as defined in [5] have a new notion of conflict freeness, they adopt Dung’s original notion of acceptability of an argument with respect to a set of arguments. Then one semantics for BAfs defined by [5] is the following:

**Definition 2** Given a BAf $=(A, D, S)$ a set $S \subseteq A$ is a d-admissible iff $S$ is conflict-free and all its elements are acceptable w.r.t. $S$. And $S$ is a d-preferred extension of BAf iff $S$ is maximal for $\subseteq$ among the d-admissible subsets of $A$.

In [5] two further semantics for BAfs are defined. They both imply that every extension is conflict-free and this will suffice for present purposes, so that they do not have to be presented here. Finally, in [6] a further notion of attack called secondary attack is defined as: if $A$ supports $B$ and $C$ attacks $B$, then $C$ (secondary-) attacks $B$.

2.3 Evidential argumentation systems

[18]’s evidential argumentation systems generalise BAfs in that both attack and support is from sets of arguments to arguments. Several differences prevent EASs from generalising BAfs in a formal sense. Formally, an EAS is a triple $(A, $D$, $S$)$, where $D \subseteq 2^A \setminus \emptyset \times A$ and $S \subseteq 2^A \cup \{\emptyset\} \times A$. Here $\emptyset$ is a special argument not in $A$ that intuitively provides support from the environment. In EASs, the attack and support relation cannot intersect: there exists no $S \subseteq 2^A$ and $A \in A$ such that both $SRAA$ and $SRAA$. Then:

**Definition 3** [Evidential support] An argument $A$ is e-supported by a set $S \subseteq A$ if:

1. $\{\emptyset\}RAA$; or
2. $\exists T \subseteq S$ such that $TRAA$ and $\forall X \subseteq T$, $X$ is e-supported by $S \setminus \{X\}$.

$S$ is a minimum e-support for $A$ if there is no $T \subseteq S$ such that $A$ is e-supported by $T$.

**Definition 4** [Evidence-supported attack] A set $S \subseteq A$ carries out an evidence-supported attack on argument $A$ iff:

1. $S'RAA$ for some $S' \subseteq S$; and
2. All elements of $S'$ are e-supported by $S$.

A supported attack by $S$ on $A$ is minimal if there is no $T \subseteq S$ such that $T$ carries out an evidence-supported attack on $A$.

**Definition 5** [Acceptability] An argument $A$ is acceptable wrt a set of arguments $S$ iff:

1. $S$ e-supports $A$; and
2. for every minimal evidence-supported attack $X$ against $A$ there exists a $T \subseteq S$ such that $TRAX$ for some $B \in X$ such that $X \setminus \{B\}$ is no longer an evidence-supported attack on $A$.

Finally, a set of arguments $S$ is conflict-free iff $\forall Y \subseteq S, \exists X \subseteq S$ such that $XRAX$. Then the notions of admissible sets and preferred extensions are defined as usual.

2.4 The ASPIC+ framework

The ASPIC+ framework [21, 15, 16] gives structure to Dung’s arguments and defeat relation. It defines arguments as inference trees formed by applying strict or defeasible inference rules to premises formulated in some logical language. Arguments can be attacked on their (non-axiom) premises and on their applications of defeasible inference rules. Some attacks succeed as defeats, which is partly determined by preferences. The acceptability status of arguments is then defined by applying any of [8] semantics for abstract argumentation frameworks to the resulting set of arguments with its defeat relation. Below the version of ASPIC+ defined in [15] is presented, more precisely, the special case with symmetric negation.

ASPIC+ is not a system but a framework for specifying systems. It defines the notion of an abstract argumentation system as a structure consisting of a logical language $L$ closed under negation, a set $R$ consisting of two subsets $R_+$ and $R_-$ of strict and defeasible inference rules, and a naming convention $n$ in $L$ for defeasible rules in order to talk about the applicability of defeasible rules in $L$. Informally, $n(r)$ is a wff in $L$ which says that $r \in R$ is applicable.

**Definition 6** [Argumentation systems] An argumentation system is a triple $AS = (L, R_+, n)$ where:

1. $L$ is a logical language closed under negation ($\neg$).
2. $R = R_+ \cup R_-$ is a set of strict ($R_+$) and defeasible ($R_-$) inference rules of the form $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$ and $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$ respectively (where $\varphi_i, \varphi$ are meta-variables ranging over wff in $L$), and $R_+ \cap R_-=\emptyset$.
3. $n$ : $R_+ \Rightarrow L$ is a naming convention for defeasible rules.

We write $\neg \varphi$ just in case $\varphi = \neg \varphi$ or $\varphi = \neg \varphi$.

**Definition 7** [Knowledge bases] A knowledge base in an $AS = (L, R, n)$ is a set $K \subseteq L$ consisting of two disjoint subsets $K_+$ (the axioms) and $K_-$ (the ordinary premises).

Arguments can be constructed step-by-step from knowledge bases by chaining inference rules into trees. In what follows, for a given argument the function $\text{Pre}$ returns all its premises, $\text{Con}$ returns its conclusion, $\text{Sub}$ returns all its sub-arguments and $\text{TopRule}$ returns the last inference rule applied in the argument.

**Definition 8** [Arguments] An argument $A$ on the basis of a knowledge base $KB$ in an argumentation system $(L, R, n)$ is:

1. $\varphi$ if $\varphi \in K$ with: $\text{Pre}(A) = \{\varphi\}$; $\text{Con}(A) = \emptyset$; $\text{Sub}(A) = \emptyset$; $\text{TopRule}(A) = \emptyset$.
2. $A_1, \ldots, A_n \Rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments such that there exists a strict/defeasible rule $\text{Con}(A_1), \ldots, \text{Con}(A_n) \Rightarrow \psi$ in $R_+ \cup R_-$. $\text{Pre}(A) = \text{Pre}(A_1) \cup \ldots \cup \text{Pre}(A_n)$, $\text{Con}(A) = \emptyset$, $\text{Sub}(A) = \text{Sub}(A_1) \cup \ldots \cup \text{Sub}(A_n) \cup \{A\}$; $\text{TopRule}(A) = \text{Con}(A_1), \ldots, \text{Con}(A_n) \Rightarrow \psi$.  

[5] use $R_{att}$ and $R_{sup}$ for the defeat and support relation and call defeat “attack”. Below “attack” in descriptions of BAfs should be read as “defeat” whenever a BAf is generated in ASPIC+.

[6] As above, “attack” in descriptions of EASs should be read as “defeat” whenever an EAS is generated in ASPIC+.
Arguments can be attacked in three ways: on their premises (undermining attack), on their conclusion (rebutter attack) or on an inference step (undercutting attack). The latter two are only possible on applications of defeasible inference rules.

**Definition 9 [Attack]** A attacks B iff A undercuts, rebuts or undermines B, where:

- A undercuts argument B (on B') iff Conc(A) = ¬n(r) and B' ∈ Sub(B) such that B's top rule r is defeasible.
- A rebuts argument B (on B') iff Conc(A) = ¬φ for some B' ∈ Sub(B) of the form B'₁, ..., B'ₙ ⇒ φ.
- Argument A undermines B (on B') iff Conc(A) = ¬φ for some B' = φ, φ ∉ Kn.

Argumentation systems plus knowledge bases form argumentation theories, which induce structured argumentation frameworks.

**Definition 10 [Structured Argumentation Frameworks]** Let AT be an argumentation theory (AS, KB). A structured argumentation framework (SAF) defined by AT, is a triple (A, C, ≤) where A is the set of all finite arguments constructed from KB in AS, ≤ is an ordering on A, and (X, Y) ∈ C iff X attacks Y.

The notion of defeat can then be defined as follows. Undercutting attacks succeed as defeats independently of preferences over arguments, since they express exceptions to defeasible inference rules. Rebutting and undermining attacks succeed only if the attacked argument is not stronger than the attacking argument (A ≤ B is defined as usual as A ≤ B and B ∉ A).

**Definition 11 [Defeat]** A defeats B iff: A undercuts B, or; A rebuts/undermines B on B' and A ≠ B'.

Abstract argumentation frameworks are then generated from argumentation theories and an argument ordering as follows:

**Definition 12 [Argumentation frameworks]** An abstract argumentation framework (AF) corresponding to a SAF = (A, C, ≤) is a pair (A, D) such that D is the defeat relation on A determined by SAF.

### 3 A simple framework for abstract support relations

In this section a simple way is proposed to add support relations between arguments to AFs, and ASPIC⁺ will be shown to instantiate it. The idea is taken from [10] and amounts to adding a binary support relation S on A to AFs with the sole additional constraint that if B supports C and A attacks B then A also attacks C ([10] also assume that the support relation is a partial order but for present purposes this assumption is not needed). [10] actually do not make this proposal to extend AFs with support relations but as part of a proposal to combine AFs with a Tarski-style consequence notion over a logical language for conclusions of arguments and to instantiate it with special cases of ASPIC⁺ and ABA.

**Definition 13 [AFs with support]** An abstract argumentation framework with support (SuppAF) is a triple (A, D, S), where A is a set of arguments, D ⊆ A × A is a binary relation of defeat and S ⊆ A × A is a binary relation of support such that if A supports B and C defeats A then C defeats B.

The semantics of SuppAFs is simply defined by choosing one of the Dung-style semantics for the corresponding pair (A, D). Thus the support relation S is only used to constrain the defeat relation D. To show that ASPIC⁺ in fact generates SuppAFs, take D to be ASPIC⁺’s defeat relation and S to be ASPIC⁺’s subargument relation between arguments. It is then immediate from Definitions 9 and 11 that ASPIC⁺’s notion of defeat satisfies [10]’s constraint on D in terms of S. This proves that SuppAFs can be instantiated with ASPIC⁺.

An equivalent reformulation of SuppAFs does make use of support relations in its semantics. In [22] ASPIC⁺ as presented above was reformulated in terms of [20]’s recursive labellings. Abstracting this reformulation to SuppAFs we obtain the following definitions. First, [22] defines a notion of p-defeat (for “Pollock-defeat”), which captures direct defeat between arguments:

**Definition 14 [p-Attack]** A p-attacks B iff A p-undercuts, p-rebuts or p-undermines B, where:

- A p-undercuts argument B iff Conc(A) = ¬n(r) and B has a defeasible top rule r.
- A p-rebuts argument B iff Conc(A) = ¬Conc(B) and B has a defeasible top rule.
- Argument A p-undermines B iff Conc(A) = ¬φ and B = φ, φ ∉ Kn.

**Definition 15 [p-Defeat]** A p-defeats B iff: A p-undercuts B, or; A p-rebuts/p-undermines B and A ≠ B.

Then [22] proves that A defeats B if A p-defeats B or a proper subargument of B. Now if the support relation of a SuppAF is taken to be ASPIC⁺’s notion of an ‘immediate’ subargument and the defeat relation of a SuppAF is taken to be p-defeat, then the following definition is equivalent to [8]’s semantics for AFs (and so for SuppAFs).

**Definition 16 [p-labellings for SuppAFs]** Let (A, D, S) be a SuppAF corresponding to a SAF = (A, D) where D is defined as p-defeat and where S is defined as (A, B) ∈ S iff B is of the form B₁, ..., Bₙ ⇒ φ and A = Bᵢ for some 1 ≤ i ≤ n. Then (In, Out) is a p-labelling of SuppAF iff In ∩ Out = Ø and for all A ∈ A it holds that:

1. A is labelled in iff:
   - (a) All arguments that p-defeat A are labelled out; and
   - (b) All B that support A are labelled in.
2. A is labelled out iff:
   - (a) A is p-defeated by an argument that is labelled in; or
   - (b) Some B that supports A is labelled out.

The notions of complete, stable, preferred and grounded labellings are defined as usual: a complete labelling is any labelling, a stable labelling labels all arguments, a preferred labelling maximises the set of arguments labelled in while a grounded labelling minimises this set. Then the corresponding notions of complete, stable, preferred and grounded extensions are defined as a set of all arguments labelled in some complete (stable, preferred, grounded) labelling.

It can be shown that the extensions defined thus for SuppAFs generated from ASPIC⁺ with p-defeat are exactly the extensions of SuppAFs as generated above from ASPIC⁺ with defeat. The proof is a straightforward generalisation of Theorem 2 of [22].
An alternative attempt to reconstruct ASPIC+ as an instance of SuppAFs is to define the support relation as follows: $A \in A$ supports $B \in A$ iff either $A$ is a subargument of $B$ or the conclusion of $A$ is a premise of $B$. However, this proposal does not work, since it cannot distinguish between the following two situations:

**Situation 1:** $A$ has premises $p$ and $q$, $B$ has conclusion $p$, $C$ has conclusion $q$, $D$ undercuts $C$.

**Situation 2:** $A$ has premise $p$ and both $B$ and $C$ have conclusion $p$, $D$ undercuts $C$.

Both situations induce the same SuppAF, in which both $B$ and $C$ support $A$ while $D$ defeats both $A$ and $C$. However, this is counterintuitive, since in the second situation $A$ should not be defeated, since its premise $p$ is still provided by an undefeated argument, namely, $B$.

In fact, this problem was already noted by Pollock [20] when he defined his notion of an inference graph, in which nodes are linked by both support and attack links. As a solution, the nodes in Pollock’s inference graphs do not simply stand for statements but encode the way in which they are derived from previous statements. In our situation 2 there would thus be two nodes for statement $p$, one as derived with argument $B$ and another as derived with argument $C$. This solution is also adopted in ASPIC+, which considers two versions of $A$, one as supported by $B$ and the other as supported by $C$.

### 4 BAFs as abstraction of ASPIC+

In this section the relation between BAFs and ASPIC+ is investigated. In fact, BAFs can in a trivial sense be instantiated with ASPIC+ since ASPIC+ generates Dung-style AFs and these are the special case of BAFs with an empty support relation. However, such an instantiation is clearly not insightful; what we would like is an instantiation with a non-empty support relation that corresponds in a meaningful way to ASPIC+’s notion of inferential support. Above we saw that a definition of support as having a premise of the supported argument as the conclusion will not work. Therefore the same definition will be used as for SuppAFs, namely as ASPIC+’s notion of a subargument.

Next, neither [10] nor ASPIC+ adopt [5]’s constraint that the attack and support relations are disjoint. It is easy to provide instantiations of ASPIC+ that violate this constraint. A simple example is with $\mathcal{L}$ a propositional language, $K_n = \{p\}$, $K_p = \emptyset$, $R_s = \emptyset$, $R_d = \{p \Rightarrow \neg p\}$. We have the following arguments:

- $A_1: p$
- $A_2: A_1 \Rightarrow \neg p$

Here $A_1$ supports $A_2$ since $A_1$ is a subargument of $A_2$ while $A_1$ also attacks and defeats $A_2$.

[5, p. 69] motivate their exclusion of such examples by saying that “... it does not seem rational to advance an argument that simultaneously attacks and supports the same other argument.” While this makes sense, it may not always be easy to detect that one argument both supports and attacks another. For this reason ASPIC+ takes an alternative approach, namely, to allow such examples and let the logic deal with them in a rational way. For instance, in ASPIC+ the above example has a unique preferred (and grounded) extension containing only $A_1$, which seems the intuitively correct outcome.

Let us now see whether BAFs can in a non-trivial sense be instantiated with ASPIC+. It is easy to see that if in BAFs only direct and secondary attacks are used to define conflict-freeness, then BAFs are equivalent to SuppAFs, since the notion of secondary attack is equivalent to [10]’s constraint on $D$. According to [17], secondary attack (and so [10]’s constraint) is suitable if support is to mean necessary support in that ‘$A$ supports $B$’ means that $B$ cannot be accepted without $A$. As also remarked in [7], ASPIC+’s subargument relation is such a relation of necessary support, since in ASPIC+ an argument cannot be in an extension if not all its subarguments are in that extension. So we have identified a realistic instantiation of [17]’s notion of necessary support.

However, things are different for variants of BAFs with supported attacks. Consider the following well-known example from the literature on nonmonotonic logic, with a propositional language and $K_n = \{q, r\}; K_p = \emptyset$; $R_s = \{r_1, r_2\}$ where:

- $r_1: q \Rightarrow p$
- $r_2: r \Rightarrow \neg p$

Read this as: quakers are typically pacifist, republicans are typically not pacifists, Richard Nixon was both a quaker and a republican. We have the following arguments:

- $A_1: q$
- $A_2: A_1 \Rightarrow p$
- $B_1: r$
- $B_2: B_1 \Rightarrow \neg p$

In ASPIC+ as reconstructed in a SuppAF we have that $A_1$ supports itself and $A_2$, $B_1$ supports itself and $B_2$ and (if the defeasible rules have equal strength) $A_2$ and $B_2$ defeat each other by successfully rebutting each other. This yields two preferred extensions:

- $E_1 = \{A_1, B_1, A_2\}$
- $E_2 = \{A_1, B_1, B_2\}$

However, in BAFs neither of these extensions is +conflict-free: in $E_1$ this is since $\{B_1\}$ set-attacks $A_2$ while in $E_2$ this is since $\{A_1\}$ set-attacks $B_2$. Instead, the preferred d-extensions in BAFs are

- $E_1 = \{A_1, A_2\}$
- $E_2 = \{B_1, B_2\}$
- $E_3 = \{A_1, B_1\}$

Thus in the BAF treatment of this example we cannot rationally accept both that Nixon was a quaker and that he was not a pacifist. This outcome shows that BAFs are not adequate as an abstraction of ASPIC+ (a similar observation is made by [13]). Moreover, the BAF outcome is arguable counterintuitive, since there is nothing inconsistent in saying both that Nixon was a quaker and that he was not a pacifist. The point is that the two generalisations in the example are defeasible, so that it can be perfectly rational to accept their antecedent but not their consequent.

Following [2], it is suggested in [6] that supported attacks do make sense for a notion of deductive support, defined as: if $A$ supports $B$ and $A$ is accepted, then $B$ must also be accepted (so if $B$ is not accepted, then $A$ cannot be accepted). Does this abstract notion of deductive support correspond to something meaningful in ASPIC+? One would expect that this is the case for classical-logic instantiations of ASPIC+ in the sense of [11], where arguments are classical subset-minimal entailments from consistent subsets of a possibly inconsistent knowledge base. As shown in [15], this can be captured in ASPIC+ by letting $\mathcal{L}$ be a propositional language, letting $R_s$ be all propositionally valid inferences and by letting $K_n$ and $R_d$ empty. The question then reduces to the question whether BAFs with supported attacks are suitable as abstraction of such instantiations of ASPIC+.

It turns out that this is not the case. Consider an example with $K_p = \{p, q, \neg(p \land q)\}$ and with all arguments of equal priority. Theorem 34 of [15] implies that each maximal consistent subset of $K_p$ is contained in a stable (and so preferred) extension. So there exists a stable extension of the SuppAF induced by this argumentation theory that contains both $p$ and $\neg(p \land q)$. However, this extension is not +conflict-free: we have the argument $p, q \Rightarrow p \land q$, which
undermines the premise argument \( \neg(p \land q) \). But since the premise argument \( p \) supports \( A \), it support-attacks \( \neg(p \land q) \). So there cannot be any \( d \)-preferred extension in BAFs that contains both \( p \) and \( \neg(p \land q) \). The problem with the abstract notion of deductive support as defined above is that it neglects that an argument \( B \) supported by an argument \( A \) can have multiple subarguments, so if \( B \) (here \( p, q \to \neg(p \land q) \)) is not accepted, one can choose to accept \( A \) (here \( p \)) and instead reject one of \( B \)'s other subarguments (here \( q \)).

5 \textbf{ASPIC} as a special case of EASs

In this section it is shown that \textbf{ASPIC} is translatable as a special case of EASs. No formal result on EASs proved in [18] depends on the constraint that their attack and support relations are disjoint, so the translation result will below be formulated for EASs without this constraint. Actually, as with BAFs a trivial translation from \textbf{ASPIC} to EASs is possible, since as noted by [18], Dung-style AFSs can be translated into EASs by letting \( q \) support all other arguments and having no further support relations, and by preserving the attack relations (now formulated from singleton sets to arguments). Since \textbf{ASPIC} generates AFSs, \textbf{ASPIC} could be translated into EASs in this way. However, this translation is clearly not very insightful. Below a more interesting translation is provided, which translates the subargument relation of \textbf{ASPIC} into support relations of EASs.

\textbf{Definition 17 [from \textbf{ASPIC} to EASs]} Let \( AF = (A, D) \) be an abstract argumentation framework corresponding to a \( SAF = (A, C, \preceq) \) induced by argumentation theory \( AT = (AS, KB) \). The evidential argumentation system corresponding to \( AF \) is defined as follows:

1. \( SR_A A \iff S = \{B\} \) and \( B \) defeats \( A \).
2. \( SR_A A \iff \begin{cases} \emptyset \iff S = \{A\}; & \text{or else} \\ S = \PrSub(A) & \text{where } \PrSub(A) = \{\text{Sub}(A) \setminus \{A\}\}. \end{cases} \)

\textbf{Lemma 1}

1. Let \( S = \text{Sub}(A) \) for some \( A \). Then any \( x \in S \) is e-supported by \( S \setminus \{x\} \).

\textit{Proof:} with induction on the structure of arguments. For the base case, suppose \( \text{Sub}(x) = \{x\} \). Then \( \{\} \models_R x \iff x \in S \setminus \{x\} \). The induction hypothesis is that for any \( y \in \PrSub(x) \), we have that \( y \) is e-supported by \( S \setminus \{y\} \). Then choose \( T \) in clause 2 of Definition 3 to be \( \PrSub(x) \). Since \( \PrSub(x) \models_R x \) we have by the induction hypothesis that this clause is satisfied.

2. A defeats \( B \) iff \( \text{Sub}(A) \) carries out a minimal e-supported attack on \( B \).

\textit{Proof:} from left to right, suppose \( A \) defeats \( B \). Then \( \{A\} \models_R B \) and \( \{A\} \subseteq \text{Sub}(A) \), so the first bullet of Definition 4 is satisfied. We next prove by induction on the structure of an argument that the second bullet is satisfied. For the base case, suppose \( \text{Sub}(A) = \{A\} \). Then \( \{\} \models_R A \iff A \) is e-supported by \( \text{Sub}(A) \). Consider next any \( A \in \text{Sub}(A) \) such that all elements of \( \text{Sub}(A) \) are e-supported by \( \text{Sub}(A) \). Note that \( \PrSub(A) \subseteq \text{Sub}(A) \) and \( \PrSub(A) \models_R A \). Then by Lemma 1(2) the second bullet is satisfied. Finally, \( \text{Sub}(A) \) is a minimal e-supported attack on \( B \) since in \textbf{ASPIC} it holds that if \( A \) defeats \( B \) and \( C \) defeats \( B \) and \( A \neq C \) then \( \text{Sub}(A) \not\subseteq \text{Sub}(C) \).

From right to left holds since \( \text{Sub}(A) \) carries out a minimal e-supported attack on \( B \), so all its elements are needed to create the attack. But then no subargument of \( A \) defeats \( B \) so \( A \) defeats \( B \).

Then the following theorem can be proved (below, if terminology for \textbf{ASPIC} or AFSs is also used in EASs, the EAS notions will be preceded with an \( e \)).

\textbf{Theorem 2} Let \( AF \) be an abstract argumentation framework corresponding to a \( SAF \) such that \( S \cap D \) in the SuppAF corresponding to \textbf{ASPIC} is empty, and let EAS correspond to \( AF \). Then \( e \) is a preferred extension of \( AF \) iff \( E \) is an e-preferred extension of \( EAS \).

\textbf{Proof:} From left to right, assume \( E \) is a preferred extension of \( AF \). First, \( E \) is conflict-free, so \( E \) is e-conflict-free by definition of \( R_e \). Next we must prove that all elements of \( E \) are e-acceptable wrt \( E \). Note first that since \textbf{ASPIC} satisfies closure under subarguments (see [15]), then by Lemma 1(1) \( E \) supports all its members. Next, since \( E \) is admissible, all its elements are acceptable wrt \( E \). Then for all \( B \) that defeat a member \( A \) of \( E \), there exists a \( C \) in \( E \) that defeats \( B \). By Lemma 1(2) there exists such a \( B \) iff \( \text{Sub}(B) \) carries out a minimal evidence-supported attack on \( A \). Then, since there exists a \( C \) that defeats a subargument of \( B \), for this \( C \) we have that \( \{C\} \models_R X \) for some \( X \in \text{Sub}(B) \), while moreover, \( \text{Sub}(B) \setminus \{X\} \) is no longer an e-supported attack on \( A \) since \( \text{Sub}(B) \) is a minimal e-supported attack on \( A \). So all elements of \( E \) are e-acceptable wrt \( E \). Then the following theorem can be proved (below, if terminology for \textbf{ASPIC} or AFSs is also used in EASs, the EAS notions will be preceded with an \( e \)).

Then from left to right, suppose \( E \) is an e-preferred extension of \( E \). First, \( E \) is conflict-free, it is also conflict free by definition of \( R_e \). Next we must prove that all elements of \( E \) are acceptable wrt \( E \). Let \( B \) defeat some \( A \) in \( E \). Then by Lemma 1(2) \( \text{Sub}(B) \) carries out a minimal e-supported attack on \( A \). Since \( E \) is e-admissible, for some \( A \in E \), we have that \( \{C\} \models_R X \) for some subargument \( B \) of \( C \). Then \( C \) defeats \( B \), so \( A \) is acceptable wrt \( E \), so \( E \) is admissible. Suppose next for contradiction that \( E \) is not maximally admissible. Then some argument \( A \) is acceptable wrt \( E \) but not in \( E \). Since \( E \) is an e-preferred extension, \( A \) is not e-acceptable wrt \( E \). Note that since \( A \) is acceptable wrt \( E \), all \( B \) that defeat \( A \) are defeated by some \( C \in E \). By Lemma 1(2) there exists such a \( B \) iff \( \text{Sub}(B) \) carries out an evidence-supported attack on \( A \). Then, since there exists a \( C \) that defeats a subargument of \( B \), for this \( C \) we have that \( \{C\} \models_R X \) for some \( X \in \text{Sub}(B) \), while moreover, \( \text{Sub}(B) \setminus \{X\} \) is no longer an e-supported attack on \( A \) since \( \text{Sub}(B) \) is a minimal e-supported attack on \( A \). So \( E \) does not e-support \( A \), otherwise \( E \) would not be maximally e-admissible. Then there exists at least one subargument \( A' \) of \( A \) of which all elements of \( \PrSub(A') \) are in \( E \) but \( A' \) is not in \( E \). Then since \( \PrSub(A') \models_R A' \), by Lemma 1(1) we have that \( E \) e-supports \( A' \). Since \( A' \) is a subargument of \( A \), we have in fact just proven that \( A' \) can be defended against all minimal evidence-supported attacks since any such attack is also a minimal e-supported attack on \( A \). So \( A' \) is e-acceptable wrt \( E \), so \( A' \) is in \( E \). Contradiction.

Given this result, the question remains what EASs offer as advantages over the simple framework for abstract support relations proposed in the present paper. One feature of EASs that was not needed to prove the correspondence with \textbf{ASPIC} is the possibility to have attacks from \textit{sets} of arguments to arguments. [18] motivate this feature with the following example.
6 Conclusion

In this paper the question was addressed whether bipolar argumentation frameworks or evidential argumentation systems can be used as an abstraction of ASPIC$^+$-style inferential support relations between arguments. This question was investigated since arguably the significance of an abstract model of argumentation depends on the range of realistic instantiations it allows. For BAFs the answer was positive for variants with only direct and secondary attacks but negative for variants with supported attacks, even for the special case of deductive support. Moreover, a simple alternative to BAFs based on an idea of [10] called SuppAFs turned out to be suitable in general as an abstraction of ASPIC$^+$-style inferential support relations. The same was proven for an abstract version of [20]’s recursive argument labellings. A question that thus remains is whether other instantiations of BAFs are possible that show their significance as a contribution to the study of argumentation. For EASs it was shown that they can be instantiated with ASPIC$^+$. This was proven for preferred semantics only, since that is the semantics on which [18] concentrate. Thus there now are two formalisms for abstract argumentation with both attack and support relations that can be instantiated in general with ASPIC$^+$, namely, EASs and SuppAFs. Clearly, EASs are more complicated as a formalism than SuppAFs. One complication is that supports in EASs are from pAFs. Clearly, EASs are more complicated as a formalism than SuppAFs and that is moreover relevant for realistic instantiations. More generally, the question can be asked what frameworks with abstract support relations offer over more concrete but still abstract frameworks with support relations defined over logical languages, such as ASPIC$^+$, assumption-based argumentation or [11]’s approach in terms of Tarskian abstract logics. But these questions have to be left for future research.

REFERENCES