Argument Accrual, *ASPIC*$^+$ and Bipolar Argumentation Frameworks

Henry PRAKKEN
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract. In this paper it is studied to which extent various semantics of bipolar argumentation frameworks are correct abstractions of four types of argument support in a version of the *ASPIC*$^+$ framework with argument accrual. Both positive and negative cases are identified. A novel positive result is that a correct abstraction for conclusion support is identified.

Keywords. Argument accrual, *ASPIC*$^+$, Bipolar Argumentation Frameworks

1. Introduction

Cohen et al. [2] studied to which extent constraints and semantics from the literature on bipolar argumentation frameworks (*BAFs*, see [1]) are suitable abstractions of four types of support relations between arguments in *ASPIC*$^+$ [4]. These four types are subargument (s-)support, conclusion (c-)support, premise (p-)support and intermediate (i-)support. Their findings were largely negative; the only positive case was about a correct abstraction of the *ASPIC*$^+$ subargument relation.

According to [2] c-support is when two arguments have the same conclusion, p-support is when the conclusion of one argument equals the premise of another, and i-support is when the conclusion of one argument equals the conclusion of a proper non-premise subargument of another argument. [6,9] called c-support *accrual* of arguments. Note that p- and i-support are in fact variants of argument accrual, since they imply c-support relations between arguments. In [9] a variant of *ASPIC*$^+$ with argument accrual was proposed. Therefore, it is interesting to repeat the investigations of [2] for this variant of *ASPIC*$^+$. Unlike [2] we will study the general case with arbitrary preference relations between arguments.

This paper is organised as follows. In Section 2 we outline the *ASPIC*$^+$ framework and the semantics of *BAFs*. In Section 3 we summarise *ASPIC*$^+$ with accrual as defined in [9]. The heart of the paper is Sections 4 and 5, in which we study for each of the four types of support to which extent the various *BAF* semantics are correct abstractions of *ASPIC*$^+$ with accrual. We conclude in Section 6.

2. Formal preliminaries

In this section we summarise the variant of the *ASPIC*$^+$ framework used in this paper and some basics of the semantics of *BAFs*. Both formalisms build on the notion of an

abstract argumentation framework (AF) [3], which is a pair $(\mathcal{A}, \mathcal{D})$, where $\mathcal{A}$ is a set of arguments and $\mathcal{D} \subseteq \mathcal{A} \times \mathcal{A}$ is a relation of defeat[1]. A **labelling** of a set $\mathcal{A}$ of arguments of an abstract argumentation framework $(\mathcal{A}, \mathcal{D})$ is any pair of non-overlapping subsets (*in*,*out*) of $\mathcal{A}$. A labelling of $\mathcal{A}$ is a *d-labelling* of $(\mathcal{A}, \mathcal{D})$ iff it satisfies:

1. an argument is *in* iff all arguments defeating it are *out*.
2. an argument is *out* iff it is defeated by an argument that is *in*.

Then *stable semantics* d-labels all arguments *in* or *out*, while *grounded semantics* minimises and *preferred semantics* maximises the set of arguments that are d-labelled *in* and *complete semantics* allows any d-labelling. The corresponding *extensions* for a semantics are the *in* sets of any labelling according to that semantics.

The ***ASPIC***$^+$ **framework** [4] defines notions of argument and defeat. In its usual formulation it directly generates abstract AFs, but for present purposes it is convenient to work with a formulation in terms of recursive labellings, inspired by similar constructs in [5]. This formulation was in [7] shown to be equivalent to the usual formulation of *ASPIC*$^+$ and it was in [9] used to model accrual of arguments. Arguments are in *ASPIC*$^+$ formed by recursively applying strict and/or defeasible rules over a logical language, starting with premises from a knowledge base. Formally:

Definition 1 [**Argumentation System**] An *argumentation system* (AS) is a tripe $AS = (\mathcal{L}, \mathcal{R}, n)$ where:

- $\mathcal{L}$ is a logical language containing a negation symbol $\neg$. We write $\psi = -\varphi$ iff $\psi = \neg\varphi$ or $\varphi = \neg\psi$.
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules of the form $\{\varphi_1, \ldots, \varphi_n\} \rightarrow \varphi$ and $\{\varphi_1, \ldots, \varphi_n\} \Rightarrow \varphi$ respectively (where φ_i, φ are meta-variables ranging over wff in $\mathcal{L}$), such that $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$.
- n is a partial function such that $n : \mathcal{R}_d \longrightarrow \mathcal{L}$.

In this paper we assume like [2] that no two different subarguments of an argument have the same conclusion.

Definition 2 [**Knowledge Bases and Arguments**] An *argument* A on the basis of a *knowledge base* $\mathcal{K}_n \cup K_p \subseteq \mathcal{L}$ (where $\mathcal{K}_n \cap \mathcal{K}_p = \emptyset$) in an argumentation system AS is a structure obtainable by applying one or more of the following steps finitely many times:

1. φ if $\varphi \in \mathcal{K}$ with: $\texttt{Prem}(A) = \{\varphi\}$, $\texttt{Conc}(A) = \varphi$, $\texttt{Sub}(A) = \{\varphi\}$, $\texttt{ImmSub}(A) =$ $\texttt{Rules}(A) = \texttt{DefRules}(A) = \emptyset$, $\texttt{TopRule}(A) =$ undefined.
2. $A_1, \ldots A_n \rightarrow\!\!\!\!/\Rightarrow \psi$ if $A_1, \ldots, A_n$ are arguments such that $\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow\!\!\!\!/\Rightarrow$ $\psi \in \mathcal{R}$ with: $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$, $\texttt{Conc}(A) = \psi$, $\texttt{Sub}(A) =$ $\texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$, $\texttt{ImmSub}(A) = \{A_1, \ldots, A_n\}$, $\texttt{Rules}(A) = \texttt{Rules}(A_1) \cup \ldots \cup \texttt{Rules}(A_n) \cup \{\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow\!\!\!\!/\Rightarrow \psi\}$, $\texttt{DefRules}(A) = \texttt{Rules}(A) \cap \mathcal{R}_d$, $\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \rightarrow\!\!\!\!/\Rightarrow \psi$.

$\mathcal{K}_n$ are the *necessary* (non-attackable) premises while $\mathcal{K}_p$ are the *ordinary* (attackable) premises. Each of the functions Func in this definition is also defined on sets of arguments $S = \{A_1, \ldots, A_n\}$ as follows: $\texttt{Func}(S) = \texttt{Func}(A_1) \cup \ldots \cup \texttt{Func}(A_n)$.

Arguments can be attacked on an ordinary premise (undermining attack), on a defeasibly derived conclusion (rebutting attack) and on a defeasible inference step (undercutting attack). Unlike in 'standard' *ASPIC*$^+$ [4], in its recursive-labelling version argu-

[1] We rename Dungs 'attack' to 'defeat' so that we can instantiate it with the *ASPIC*$^+$ defeat relation.

ments can only be attacked and defeated on their final conclusion or inference; the recursion takes care of attack on a subargument. Accordingly, in the following definitions several notions are preceded by a p (for 'Pollock') to distinguish them from the original notions of *ASPIC*$^+$ in e.g. [4]. As usual, $A \prec B$ is defined as $A \preceq B$ and $B \not\preceq A$ while $A \approx B$ is defined as $A \preceq B$ and $B \preceq A$.

Definition 3 [**p-attack and p-defeat**] A *p-attacks* B iff A *p-undercuts*, *p-rebuts* or *p-undermines* B, where:

- A *p-undercuts* B iff $\texttt{Conc}(A) = -n(r)$ and B has a defeasible top rule r.]
- A *p-rebuts* B iff $\texttt{Conc}(A) = -\texttt{Conc}(B)$ and B has a defeasible top rule.
- A *p-undermines* B iff $\texttt{Conc}(A) = -\varphi$ and B = φ, where $\varphi \notin \mathcal{K}_n$.

An argument is *non-attackable* iff it has no ordinary premises and applies no defeasible rules. Finally, A *p-defeats* B iff: A p-undercuts B, or else; A p-undermines or p-rebuts B and $A \not\prec B$.

Definition 4

1. A *p-structured argumentation framework (pSAF)* defined by pair $(AS, \mathcal{K})$ is a triple $\langle \mathcal{A}, \mathcal{C}^p, \preceq \rangle$ where $\mathcal{A}$ is the set of all arguments constructed on the basis of $\mathcal{K}$ in AS, $\preceq$ is an ordering on $\mathcal{A}$, and $(X, Y) \in \mathcal{C}^p$ iff X p-attacks Y.
2. A *p-abstract argumentation framework (pAF) corresponding to a pSAF* = $\langle \mathcal{A}, \mathcal{C}^p, \preceq \rangle$ is a pair $(\mathcal{A}, D)$ such that D is the p-defeat relation on $\mathcal{A}$ determined by *pSAF*.

Argument evaluation is then recursively defined as follows:

Definition 5 [p-labellings.]

1. A pair of subsets (*in*,*out*) of $\mathcal{A}$ is a *p-labelling* of a *pAF* iff *in* $\cap$ *out* $= \emptyset$ and for all $A \in \mathcal{A}_{pAF}$ it holds that:
   (a) A is labelled *in* iff:
      i. All arguments that p-defeat A are labelled *out*; and
      ii. All members of $\texttt{ImmSub}(A)$ are labelled *in*; and

   (b) A is labelled *out* iff:
      i. A is p-defeated by an argument that is labelled *in*; or
      ii. A member of $\texttt{ImmSub}(A)$ is labelled *out*.

Then *stable semantics* p-labels all arguments, while *grounded semantics* minimises and *preferred semantics* maximises the set of arguments that are p-labelled *in* and *complete semantics* allows any p-labelling.

A link with standard *ASPIC*$^+$ can be made by observing that A defeats B in standard *ASPIC*$^+$ just in case A p-defeats B or a proper subargument of B. Given this link, [7] showed that the d-labellings of the AF corresponding to a *p-SAF* given the defeat relation coincide with the p-labellings of the *pSAF* given the p-defeat relation.

Bipolar argumentation frameworks ($BAFs$) [1] are a triple $(\mathcal{A}, \mathcal{D}, \mathcal{S})$ where $\mathcal{A}$ and $\mathcal{D}$ are defined as above and $\mathcal{S}$ is a support relation on $\mathcal{A}$. A *sequence of supports* for argument B by argument A is a sequence $A\mathcal{S}B_1, \ldots \mathcal{S}B_n\mathcal{S}B$. The semantics of $BAFs$ can be defined in terms of constraints on the defeat relation given sets of defeat and support relations between arguments. Arguments are then evaluated by applying a given

Dung-style semantics to AFs that contain all additional defeats induced by these constraints. Cohen et al. [2] consider the following constraints. Given a $BAF = (\mathcal{A}, \mathcal{D}, \mathcal{S})$:

1. there is a *supported defeat* from A to B iff there exists an argument C such that there is a sequence of supports from A to C and $(C, B) \in \mathcal{D}$;
2. there is a *secondary defeat* from A to B iff there exists an argument C such that there is a sequence of supports from C to B and $(A, C) \in \mathcal{D}$;
3. there is an *extended defeat* from A to B iff there exists an argument C such that there is a sequence of supports from C to A and $(C, B) \in \mathcal{D}$;
4. there is a *mediated defeat* from A to B iff there exists an argument C such that there is a sequence of supports from B to C and $(A, C) \in \mathcal{D}$.

Cohen et al. [2] define four types of support within ***ASPIC***$^+$:

Definition 6 [**Types of support in *ASPIC***$^+$] For any pair $(AS, \mathcal{K})$ and any pair of arguments A, B on the basis of $\mathcal{K}$ in AS:

1. A *subargument-supports* B iff A is a subargument of B and $A \neq B$;
2. A *conclusion-supports* B iff $\texttt{Conc}(A) = \texttt{Conc}(B)$ and $A \neq B$;
3. A *premise-supports* B iff $\texttt{Conc}(A) \in \texttt{Prem}(B)$ and $A \neq B$;
4. A *intermediate-supports* B iff for some $B' \in \texttt{Sub}(B)$ it holds that $\texttt{Conc}(B') = \texttt{Conc}(A)$ and $\texttt{Conc}(B') \neq \texttt{Conc}(B)$ and $\texttt{Conc}(B') \notin \texttt{Prem}(B)$.

Cohen et al. then investigate for various 'semantics', i.e., various combinations of constraints on ***BAFs*** whether their four notions of support in ***ASPIC***$^+$ can be related to these three BAF semantics. For each $AF = (\mathcal{A}, \mathcal{D})$ induced by an ***ASPIC***$^+$ instantiation they first consider the $BAF = (\mathcal{A}, \mathcal{D}, \mathcal{S}_s)$ where $\mathcal{S}_s$ is the support relation on $\mathcal{A}$ according to ***ASPIC***$^+$ support type s. Then for each semantics x they consider $BAF_x = (\mathcal{A}, \mathcal{D}_x, \mathcal{S}_s)$, where $\mathcal{D}_x$ adds to $\mathcal{D}$ all defeats induced by the constraints of semantics x. They then compare for each ***ASPIC***$^+$-induced $BAF = (\mathcal{A}, \mathcal{D}, \mathcal{S}_s)$ and each corresponding $BAF_x = (\mathcal{A}, \mathcal{D}_x, \mathcal{S}_s)$ the AFs $(\mathcal{A}, \mathcal{D})$ and $(\mathcal{A}, \mathcal{D}_x)$. Given a semantics y for AFs, the semantics x is an abstraction of support type s just in case $(\mathcal{A}, \mathcal{D})$ and $(\mathcal{A}, \mathcal{D}_x)$ have the same extensions according to y. Cohen et al.'s findings on this question are largely negative. They identify only one full correspondence, between ***ASPIC***$^+$ proper-subargument support and $BAFs$ with secondary and extended defeats.

3. Accrual as modelled by [9]

We first recall the three principles of accrual proposed in [6]. (1) An accrual is sometimes weaker than its accruing elements, due to the possibility that accruing reasons are not independent. (2) For the same reason an accrual makes its elements inapplicable. (3) Flawed arguments should not accrue. If an argument is based on grounds that are refuted, it should not enter into the accrual.

We next summarise the accrual model of [9][2]. For motivations of the design decisions see that paper. A first key idea is that arguments for the same conclusion are collected in a so-called accrual set. Such sets are defined relative to a labelling of the set $\mathcal{A}$ of arguments, to be able to express that only arguments of which no immediate subargu-

[2]This will take much space but this is inevitable given the relative novelty of the proposal of [9].

ment is *out* and no undercutter is *in* enter the accrual set. Note that the labelling referred to in the next definition does not have to satisfy the contraints of a d-labelling.

Definition 7 [**Accrual sets**] Given a labelling $l = (in,out)$ of $\mathcal{A}$, a nonempty set $S \subseteq \mathcal{A}$ is an *accrual set* of $\varphi \in \mathcal{L}$, denoted as $s_l(\varphi)$ iff it satisfies the following conditions:

1. $\texttt{Conc}(A) = \varphi$ for all $A \in S$;
2. if $A \in s_l(\varphi)$, then no p-undercutter of A is *in* and no member of $\texttt{ImmSub}(A)$ is *out*;
3. if all p-undercutters of A are *out* and all members of $\texttt{ImmSub}(A)$ are *in*, then $A \in s_l(\varphi)$.

Given a labelling $l = (in,out)$ of $\mathcal{A}$, any argument A of which no undercutter is *in* and no argument in $\texttt{ImmSub}(A)$ is *out* is *weakly applicable in l* and any argument A of which all undercutters are *out* and all arguments in $\texttt{ImmSub}(A)$ are *in* is *strongly applicable in l*. Then any accrual set of φ contains all strongly applicable arguments for φ (by condition 2) plus possibly some weakly applicable arguments for φ (by condition 1).

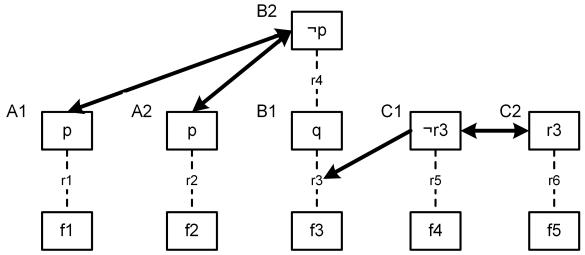

Figure 1. Running example

Example 1 [**running example**] Let $\mathcal{K} = \{f_1, f_2, f_3, f_4, f_5\}$, $\mathcal{R}_s = \emptyset$ and $\mathcal{R}_d = \{f_1 \Rightarrow_{r1} p; f_2 \Rightarrow_{r2} p; f_3 \Rightarrow_{r3} q; q \Rightarrow_{r4} \neg p; f_4 \Rightarrow_{r5} \neg r_3; f_5 \Rightarrow_{r6} r_3\}$ (with the rule names attached to the arrows). The following arguments can be constructed (leaving the premise arguments implicit).

A_1: $f_1 \Rightarrow p$ A_2: $f_2 \Rightarrow p$ B_1: $f_3 \Rightarrow q$ B_2: $B_1 \Rightarrow \neg p$
C_1: $f_4 \Rightarrow \neg r_3$ C_2: $f_5 \Rightarrow r_3$

Figure 1 displays these arguments and their attack relations: C_1 undercuts B_1, while all symmetric defeat relations are rebuttals. Let l_1 be the empty labelling. Then p has three accrual sets $\{A_1\}$, $\{A_2\}$ and $\{A_1, A_2\}$. Next let l_2 be such that f_1, f_2 and C_1 are *in* while C_2, B_1 and B_2 are *out* and all other arguments are undecided. Then $\{A_1, A_2\}$ is the only accrual set for p while $\neg p$ does not have an accrual set.

The argument preference relation $\preceq$ is now redefined as a relation on the powerset of $\mathcal{A}$ satisfying some properties in order to properly handle cases with unattackable arguments. This induces the notion of an accrual argumentation framework:

Definition 8 [**l-Argumentation frameworks**] An *accrual argumentation framework* (*aSAF*) defined by an $AT = (AS, \mathcal{K})$ is a tuple $\langle \mathcal{A}, \mathcal{C}^p, \preceq \rangle_{AT}$ where

1. $\mathcal{A}$ is the set of all arguments on the basis of $\mathcal{K}$ in AS;

2. $\mathcal{C}^p$ is the p-attack relation on $\mathcal{A}$;
3. $\preceq$ is a reflexive binary relation on $2^{\mathcal{A}}$ satisfying:
   (a) $S \preceq S'$ for any S' that contains a strict- and firm argument.
   (b) $S \not\prec S'$ for any S that contains a strict- and firm argument.

Next the notion of defeat is made relative to a labelling, so that in case of rebutting arguments only those arguments are compared that have no undercutters that are *in* and immediate subarguments that are *out*. This helps to satisfy the third principle of accrual.

Definition 9 [**l-defeat**] Given a labelling $l = (in,out)$ of the set $\mathcal{A}$ of an $aSAF$, argument A *l-pdefeats* argument B iff:

1. A p-undercuts B; or
2. A p-undermines or p-rebuts B and for some $s_l(\texttt{Conc}(A))$ including A and some $s_l(\texttt{Conc}(B))$ including B it holds that $s_l(\texttt{Conc}(A)) \not\prec s_l(\texttt{Conc}(B))$.

Note that if there is no accrual set for a conclusion of a rebutted argument, then the argument will be *out* anyway, so its defeat relations are irrelevant.

Example 2 [**running example ct'd**] Suppose first that $\{A_1\} \prec \{B_2\}$, $\{A_2\} \prec \{B_2\}$ and $\{B_2\} \prec \{A_1, A_2\}$. Assume first a labelling l in which all arguments are undecided. Then p and $\neg p$ have the same accrual sets as for l_1 in Example 1. Then both A_1 and A_2 l-pdefeat B_2 since $\{B_2\} \prec \{A_1, A_2\}$ but B_2 also l-pdefeats A_1 since $\{A_1\} \prec \{B_2\}$ and B_2 l-pdefeats A_2 since $\{A_2\} \prec \{B_2\}$. Assume next a labelling l' in which all premise arguments are *in* while the other arguments are all undecided. Then $\{A_1, A_2\}$ is the only accrual set for p, so now both A_1 and A_2 strictly l'-pdefeat B_2 since $\{B_2\} \prec \{A_1, A_2\}$. Assume next that the preference relations between all above argument sets are reversed and consider again labelling l'. Then B_2 strictly l-pdefeats both A_1 and A_2 because of $\{A_1, A_2\} \prec \{B_2\}$, even though both $\{B_2\} \prec \{A_1\}$ and $\{B_2\} \prec \{A_2\}$. The reason is that $\{A_1, A_2\}$ is still the only accrual set for p.

Finally, a function is defined on labellings and the new semantics are defined in terms of the fixpoints of this function.

Definition 10 [**l-labellings**] The *characteristic function* F of an $aSAF$ is a total function on the set of all labellings $l(in,out)$ of $\mathcal{A}$ that returns a labelling $l' = (in',out')$ of $\mathcal{A}$ according to the following conditions: for all $A \in \mathcal{A}$ it holds that:

1. $A \in$ *in*' iff:
   (a) all arguments that l-pdefeat A are in *out*; and
   (b) all members of $\texttt{ImmSub}(A)$ are in *in*; and
2. $A \in$ *out*' iff:
   (a) A is l-pdefeated by an argument in *in* or
   (b) a member of $\texttt{ImmSub}(A)$ is in *out*.

An *l-labelling* of an $aSAF$ is any fixpoint of F.

Example 3 [**running example ct'd**] Consider first the initial argument ordering of Example 2 plus $\{C_1\} \approx \{C_2\}$ and consider the empty labelling l. Then $F(l) = l'$ makes all premise arguments *in* and leaves the other arguments undecided. Then $l'' = F(l')$ also makes A_1 and A_2 in since they have no l'-pdefeaters since they both strictly l'-pdefeat

B_2. So B_2 is *out* in l'' while the other arguments that were undecided in l' are still undecided in l'. Then $F(l'') = l''$ so l'' is the grounded labelling. There are two preferred labellings: the one extends l'' by making C_2 and B_1 *in* and C_1 *out* while the other extends l'' by making C_1 *in* and C_2 and B_1 *out*.

Consider next the second argument ordering of Example 2 plus $\{C_1\} \approx \{C_2\}$. Then the grounded labelling equals l' since, although B_2 strictly l'-pdefeats both A_1 and A_2, its subargument B_1 is undecided. There are two preferred labellings. The first extends l' by making C_2 and B_1 *in* and C_1 *out* and therefore makes B_2 *in* and both A_1 and A_2 *out*. The second makes C_1 *in* and C_2 and B_1 and B_2 *out* so it also makes A_1 and A_2 *in*.

In [9] the F operator, which always produces a d-labelling, is proven to be monotonic, which motivates the following definitions The *grounded l-labelling* is the smallest l-labelling, a *complete l-labelling* is any l-labelling, a *stable l-labelling* is a total l-labelling and a *preferred l-labelling* is any maximal l-labelling. By monotonicity of F the original relations between these semantics are then preserved, as well as the guarantee of existence of complete and preferred l-labellings.

4. Abstractions with d-defeat

We now investigate which *BAF* semantics are correct abstractions of the four kinds of support in *ASPIC*$^+$ with accrual. In this section we do this for d-defeats, both in general and for the special case with no undercutters. In particular, given an arbitrary *BAF*$_x$ induced by an AF corresponding to an *aSAF* and an l-labelling l[3], we investigate for each of the four constraints on AF given *BAF*$_x$ whether the extensions of the AF_x resulting from applying the constraints to AF equal the extensions of AF. Note that the *BAF* constraints can add defeats to an AF that do not satisfy the definitions of defeat in *ASPIC*$^+$ with accrual. The results will be presented semiformally for reasons of space.

In *ASPIC*$^+$ with accrual defeat is relative to a labelling, so AFs and $BAFs$ must now also be relative to a labelling. We say that A *d-attacks* B iff A p-attacks some subargument of B, and for a given labelling l, A *l-ddefeats* B iff A l-pdefeats some subargument of B. Then an *adSAF* of an *aSAF* $= (\mathcal{A}, \mathcal{C}^p, \preceq)$ is a triple $(\mathcal{A}, \mathcal{C}, \preceq)$ where $\mathcal{C}$ is the d-attack relation on $\mathcal{A}$. Finally, $(\mathcal{A}, \mathcal{D})$ is the AF corresponding to an *aSAF* $= (\mathcal{A}, \mathcal{C}^p, \preceq)$ and a labelling l of $\mathcal{A}$ iff $\mathcal{D}$ is the l-ddefeat relation according to $\mathcal{C}$, $\preceq$ and l, and for any support relation x, *BAF*$_x = (\mathcal{A}, \mathcal{D}, S_x)$ is the bipolar framework induced by AF by adding to it all x supports.

Definition 11 Consider an *aSAF* $= (\mathcal{A}, \mathcal{C}^p, \preceq)$ with labelling l and let A and B be arguments from *aSAF*.

1. A d-attacks B iff A p-attacks some subargument of B;
2. A l-ddefeats B iff A l-pdefeats some subargument of B;
3. $(\mathcal{A}, \mathcal{D})$ is the AF corresponding to an *aSAF* with labelling l iff $\mathcal{D}$ is the l-ddefeat relation induced by the l-pdefeat relation generated by *aSAF* and l.

Then we can prove the following lemma.

[3]From now on we only consider l-labellings and therefore omit the prefix l.

Lemma 4 Consider a wff φ, a set S of arguments such that $\texttt{Conc}(A) = \varphi$ for every $A \in S$ and a labelling l.

1. If each member of S is in some l-accrual set of φ, then
   (a) there exists an l-accrual set of φ that contains all members of S;
   (b) every l-pdefeater that p-rebuts a member of S also l-pdefeats every attackable member of S. This does not hold in general for undercutting defeaters.
   (c) If a member of S l-pdefeats A then all members of S l-pdefeat A.
2. If $A \notin S$ then
   (a) A is *out*;
   (b) A is not l-pdefeated.

We next investigate the correctness of semantics as abstractions for all four kinds of support, each time assuming a given AF corresponding to an *aSAF* and a labelling l.

4.1. Subargument support

The combination of secondary and extended defeats is a correct abstraction of s-support, since secondary defeats are implicit in the AF while adding extended defeats does not have an effect. First, the variant of Proposition 1 of [7] for l-pdefeat and l-ddefeat can be proven as in [7] since all current defeats are relative to the same labelling. Then the proof of [2, pp. 95–96] directly applies.

4.2. Conclusion support

We next investigate all criteria for c-support. Below we consider without loss of generality the special case of A_1 and A_2 for φ so A_1 c-supports A_2 and vice versa, while B is some arbitrary argument.

Secondary defeats (if you defeat the supporter, you defeat the supported). If B l-ddefeats A_1 by l-pdefeating a proper subargument of A_1, then clearly adding a defeater from B to A_2 can change the labellings and extensions.

If there are no undercutters, assume A_1 is not in an l-accrual set for φ, so A_1 is *out*, and let B l-ddefeat A_1 by l-pdefeating a proper subargument of A_1. Then if A_2 is *in*, adding a defeater from B to A_2 can change the extensions.

Extended defeats (if the supporter defeats you, then the supported defeats you). Suppose A_1 l-ddefeats B. Then A_1 l-pdefeats some subargument B' of B. Suppose A_1 is in an l-accrual set for φ. If A_2 is also in an l-accrual set for φ then they are in the same l-accrual set and by Lemma 4 A_2 l-defeats B' so the defeat from A_2 to B' is already in AF. If A_2 is not in any l-accrual set for φ then A_2 is *out* by Lemma 4, so adding an extended defeat from A_2 to B' has no effect on the extensions.

Supported defeats (if the supported defeats you, then the supporter defeats you). Suppose A_2 l-ddefeats B. Then A_2 l-pdefeats some subargument B' of B. Then A_2 is in some l-accrual set for φ. If A_1 also is in some l-accrual set for φ then A_1 also l-defeats B' so the defeat from A_1 to B' is already in AF. Otherwise A_1 is *out* and adding a defeat from A_1 to B' has no effect on the extensions.

Mediated defeats (if you defeat the supported, then you defeat the supporter). This can change extensions since defeat can be by way of p-undercutting. With rebuttals there are obvious examples with l-pdeaters of proper subarguments that change the extensions.

So for c-support only adding extended and supported defeaters does not change the extensions, since adding such defeats has no effect. This result implies that the combination of extended and supported defeat can be regarded as a correct abstraction of c-support.

4.3. Premise support

With p-support we have that if $A_2 \in \mathcal{K}$ then A_1 p-supports A_2 iff A_1 c-supports A_2, so the situation is as with c-support. Otherwise:

Secondary defeats. There are clear examples where the extensions change. For example, suppose B l-ddefeats A_1 by p-undercutting a proper subargument of A_1. Then B does not have to l-ddefeat A_2. In cases where it does not, clearly adding a defeat from B to A_2 can change the labellings and extensions.

Even without undercutters there are such examples. Suppose A_1 p-supports A_2 and B l-ddefeats A_1 by l-pdefeating a proper subargument of A_1, and suppose A_2 is *in*. Then adding a defeat from B to A_2 can change the extensions.

Extended defeats. There are clear examples where the extensions change. Suppose A_2 is a non-premise argument and A_1 symmetrically l-pdefeats B by p-rebutting it. Then adding a defeat from A_2 to B can change the extensions

Supported defeats. There are similar examples as for extended defeat. Suppose both A_1 and A_2 are non-premise arguments and A_2 symmetrically l-pdefeats B by p-rebutting it. Then adding a defeat from A_1 to B can change the extensions.

Mediated defeats. There are obvious examples with undercutters where the extensions can change, while with no undercutters there are similar examples as for supported and extended defeats.

So with p-support all kinds of defeat can change the extensions, both in general and for the special case with no undercutters.

4.4. Intermediate support

With i-support the supported conclusion must be of a non-premise proper subargument, so the supporter and the subargument on which the supported is supported are in a c-support relation.

Secondary defeats. There are similar examples as for c-support, showing that the extensions can change both in general and for the special case with no undercutters.

Extended defeats. Suppose A_2 for conclusion φ is extended to an argument A_3 for conclusion ψ unrelated to φ, let A_1 for φ i-support A_3 and let A_2 symmetrically l-pdefeat B by p-rebutting it. Then adding a defeat from A_3 to B can change the extensions.

Supported defeats. Change the example for extended defeat to the effect that now A_3 symmetrically l-pdefeats B. Then adding a defeat from A_2 to B can change the extensions.

Mediated defeats. Obvious counterexamples with undercutters and with rebuttals of proper subarguments of A_1.

So for i-support all kinds of defeat can change the extensions, both in general and for the special case with no undercutters.

Concluding Section 4, for s-support, the combination of secondary and extended defeat is, just as in [2], a correct abstraction. For the other three types of support the only positive result is that extended and supported defeats are, both individually and together, a correct abstraction of c-support.

5. Abstractions with p-defeat

Given a *BAF*$_x$ induced by an AF corresponding to an *aSAF* and a labelling l, we now consider its p-defeats only when determining which defeats have to be added. Moreover, we again investigate the special case with no undercutters.

5.1. Conclusion support

Secondary defeats. For p-undercutters there are obvious cases where the extensions change. If there are no p-undercutters then if A_1 is not in an l-accrual set for φ, then A_1 is *out*, so B cannot l-pdefeat A_1 by Lemma 4. If A_1 is in an l-accrual set for φ, then if A_2 is not, A_2 is *out* so adding a defeat from B to A_2 has no effect on the extensions. If A_2 is in an l-accrual set, then by Lemma 4 A_1 and A_2 are in the same l-accrual set, so if B l-pdefeats A_1 then B also l-pdefeats A_2 (recall there are by assumption no undercutters), so the defeat from B to A_2 is already in AF. So for undermining or rebutting p-defeat nothing changes in the extensions.

Extended defeats. Since this changes no extensions for l-ddefeats, the same holds for the special case of l-pdefeats.

Supported defeats. As for extended defeats.

Mediated defeats This can still change extensions in case of p-undercutters since the example for l-ddefeat still holds. With no p-undercutters the same line of reasoning applies as for secondary defeats, so this is a special case in which there is no change in extensions.

So for c-support in general the same holds as for d-defeat but we identified some special cases with no p-undercutters in which nothing changes in the extensions.

5.2. Premise support

Recall that for p-support we have that if $A_2 \in \mathcal{K}$ then A_1 p-supports A_2 iff A_1 c-supports A_2, so the situation is as with c-support with d-defeat. Otherwise:

Secondary defeats. In general the situation is the same as for d-defeat since there are still obvious examples with p-undercutters where the extensions can change. Suppose B l-pdefeats A_1 by p-undercutting A_1. Then B does not have to l-ddefeat A_2. In cases where it does not, clearly adding a defeat from B to A_2 can change the labellings and extensions. Even without p-undercutters there are such examples. Suppose A_1 p-supports A_2 and B l-pdefeats A_1. Then if A_1 supports A_2 on a necessary premise, B does not l-pdefeat A_2, so adding such a defeat in AF can change the extensions. By contrast, if A_1 supports A_2 on an ordinary premise, then l-pdefeating A_1 implies l-pdefeating A_2 by Lemma 4. So the case with no necessary premises is a special case in any added secondary defeat is already in AF.

Extended defeats. The situation is the same as for d-defeat, even with no p-undercutters: adding extended defeats can change extensions.

Supported defeats. The same holds as for extended defeats, even with no undercutters.

Mediated defeats. The same holds as for extended and supported defeats, even with no undercutters.

So with p-support all kinds of defeat change something, except in two cases: (1) if the supported argument is a premise argument, in which case we have a special case of c-support, and (2) with no p-undercutters and no necessary premises.

5.3. Intermediate support

Secondary defeats. Assume that $\texttt{Conc}(A_1) = \texttt{Conc}(A_2') = \varphi$ for some proper non-premise subargument A_2' of A_2. For p-undercutters there are obvious cases where the extensions change. If there are no p-undercutters then if A_1 is not in an l-accrual set of φ, then A_1 is *out*, so B cannot l-pdefeat A_1 by Lemma 4. If A_1 is in an l-accrual set of φ, then if A_2 is not, A_2 is *out* so adding a defeat from B to A_2 does not have an effect on the extensions. If A_2 is in an l-accrual set of φ, then by Lemma 4 A_1 and A_2' are in the same l-accrual set, so if B l-pdefeats A_1 then B also l-pdefeats A_2' (recall there are by assumption no undercutters). Then the defeat from B to A_2 is already in AF.

Extended defeats. Suppose A_1 l-pdefeats B but is itself *out* because of a defeat on a proper subargument. Then adding a defeat from A_2 to B can change the extensions, regardless of the type of l-pdefeat.

Supported defeats. Assume that $\texttt{Conc}(A_2) = \texttt{Conc}(A_1') = \varphi$ for some proper non-premise subargument A_1' of A_1 and assume that A_1 symmetrically l-pdefeats B by p-rebutting it. Then adding an asymmetric defeat from A_2 to B can change can change the extensions.

Mediated defeats. With p-undercutters clearly extensions can change. Without undercutters, assume that $\texttt{Conc}(A_2) = \texttt{Conc}(A_1') = \varphi$ for some proper non-premise subargument A_1' of A_1. Then an l-pdefeater of A_1 has by minimality of arguments a different conclusion than A_1' B so cannot l-pdefeat A_2. So adding a defeat from B to A_2 can change the extensions.

In sum, for i-support all constraints can change the extensions in general. Without p-undercutters there is, as for c- and p-support, a special case with secondary defeats with no change in extensions holds. Like for p- but unlike c-support there are no further special cases.

Concluding Section 5, for the general case with l-pdefeat the same holds as for l-ddefeat. For the case without p-undercutters some special cases without extension change were identified for c-support with secondary and mediated defeats, for p-support with secondary defeats and all premises ordinary, and for i-support with secondary defeats.

6. Conclusion

In this paper we investigated to which extent the *BAF* constraints and semantics studied in [2] for *ASPIC*$^+$ are for *ASPIC*$^+$ with accrual as defined in [9] correct abstractions of the four types of argument support defined by [2]. We summarise the results in Table 1. The results are similar but not identical to those of [2]. In particular, our general results on c-support are not obtained by [2]. Moreover, while we report on special cases with p-defeat, [2] instead report special cases in terms of the syntax of arguments. Finally, unlike them, we assume arbitrary preference relations between arguments.

An important result is that we have identified a semantics for c-support, namely, the combination of extended and supported defeat. Our results arguably show that it is best

Support type	**Correct abstractions for d-defeat**	**Correct abstractions for p-defeat**	**Special cases with p-defeat and no undercutters**
Subargument	Secondary defeat Extended defeat	Secondary defeat Extended defeat	
Conclusion	Extended defeat Supported defeat	Extended defeat Supported defeat	Secondary defeat Mediated defeat
Premise			Secondary defeat (all premises ordinary)
Intermediate			Secondary defeat

Table 1. Overview of results on correctness of abstractions.

to regard p- and i-support as subsumed by c-support rather than as independent types of support. In this regard it is relevant that, unlike for s- and c-support, defeating a p- or i-supporter cannot change the status of the supported argument. However, this only holds for static settings as in [2] and this paper, with a given AF generated by some structured account of argumentation. In dynamic settings of, for instance, debates, p- and i-support can have independent value as debate moves that support arguments put forward by others. One view on such debate moves, explored in [8] is that they change the supported argument.

References

[1] C. Cayrol and M.-C. Lagasquie-Schiex. Bipolar abstract argumentation systems. In I. Rahwan and G.R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 65–84. Springer, Berlin, 2009.

[2] A. Cohen, S. Parsons, E. Sklar, and P. McBurney. A characterization of types of support between structured arguments and their relationship with support in abstract argumentation. *International Journal of Approximate Reasoning*, 94:76–104, 2018.

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

[4] S. Modgil and H. Prakken. Abstract rule-based argumentation. In P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, editors, *Handbook of Formal Argumentation*, volume 1, pages 286–361. College Publications, London, 2018.

[5] J.L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person*. MIT Press, Cambridge, MA, 1995.

[6] H. Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, pages 85–94, New York, 2005. ACM Press.

[7] H. Prakken. Relating ways to instantiate abstract argumentation frameworks. In K.D. Atkinson, H. Prakken, and A.Z. Wyner, editors, *From Knowledge Representation to Argumentation in AI, Law and Policy Making. A Festschrift in Honour of Trevor Bench-Capon on the Occasion of his 60th Birthday*, pages 167–189. College Publications, London, 2013.

[8] H. Prakken. Modelling support relations between arguments in debates. In C.I. Chesñevar et al., editor, *Argumentation-based Proofs of Endearment. Essays in Honor of Guillermo R. Simari on the Occasion of his 70th Birthday*, pages 349–365. College Publications, London, 2018.

[9] H. Prakken. Modelling accrual of arguments in ASPIC+. In *Proceedings of the 17th International Conference on Artificial Intelligence and Law*, pages 103–112, New York, 2019. ACM Press.