Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220163

Strongly Accepting Subframeworks: Connecting Abstract and Structured Argumentation[1]

Markus ULBRICHT [a] and Johannes P. WALLNER [b]

[a] *Department of Computer Science, Leipzig University, Germany*
[b] *Institute of Software Technology, Graz University of Technology, Austria*
ORCiD ID: Markus Ulbricht https://orcid.org/0000-0002-0773-7510, Johannes P. Wallner https://orcid.org/0000-0002-3051-1966

Abstract. Computational argumentation is primed to strengthen the current hot research field of Explainable Artificial Intelligence (XAI), e.g., by dialectical approaches. In this paper, we extend and discuss a recently proposed approach of so-called strong acceptance on abstract argumentation that aims to support explaining argumentative acceptance. Our goal is to push these results into the realm of structured argumentation. In this setting, a knowledge base induces an abstract argumentation framework (AF) via instantiation. We investigate how and under which conditions it is possible to transfer results regarding strong acceptance between the given knowledge base and the induced AF. To this end we consider generic functions formalizing the interaction of the AF and the knowledge base. This approach helps us to infer rather general results making basic assumptions rather than dealing with the technical details of several structured argumentation formalisms. Along the way, we apply our techniques to the concrete approach of assumption-based argumentation (ABA) which constitutes one of the primal structured argumentation formalisms.

Keywords. Structured argumentation, Assumption-based argumentation, Strong acceptance

1. Introduction

Computational argumentation is a thriving research area within the broader field of knowledge representation and reasoning and the landscape of AI research [1]. With application avenues in, e.g., legal and medical research [2], a key contribution of computational argumentation are ways of specifying argument structures and argumentative acceptance forming the basis for, e.g., automated argumentative reasoning. Central to these formalizations are prescribed workflows: from a knowledge base argument structures are instantiated, together with inter-argument relations, upon which argumentation semantics define criteria of acceptance [3,4,5,6]. Importantly it was shown that after instantia-

[1]This work was supported by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig

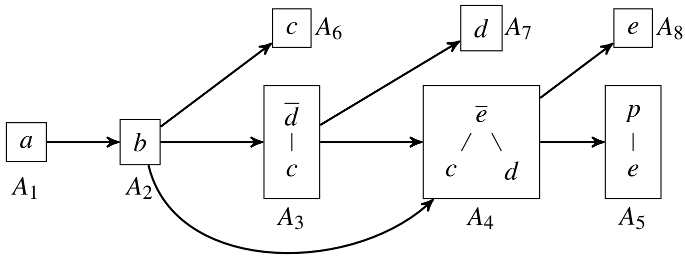


Figure 1. Arguments $\{A_3, A_5\}$ are strongly accepting p on the AF side. On the KB side, $\{(\overline{d} \leftarrow c, e), (p \leftarrow e), e\}$ is strongly accepting p.

tion, an abstract view of arguments and their relations suffices for deriving acceptance of arguments for several use cases.

Computational argumentation is actively contributing to the area of explainable artificial intelligence (XAI) [7], e.g., by providing dialectical grounds of acceptance or rejection of arguments and claims. Several approaches that support explainability arising from argumentation have been proposed and studied. We focus here on ways of supporting explanations by investigating what arguments, or parts of the knowledge base, are sufficient to show acceptance of a target conclusion or argument.

In monotonic formalisms, a common way of looking at parts that entail acceptance is to look at minimal parts (of a knowledge base) that entail the result. In non-monotonic approaches, such as virtually all approaches to argumentation, an adapted notion was presented in order to account for the fact that parts of a knowledge base might entail a certain claim, but as a whole it does not. Formally, given some set B, we can only be sure that a subset B' suffices to entail a certain piece of information, whenever this is the case for each B'' with $B' \subseteq B'' \subseteq B$.

Recently, these approaches were extended to the field of abstract argumentation [8,9,10]. However, as observed from various other aspects [11,12], connecting results from abstract to the non-abstract view is not always immediate. Consider a simple example in the structured argumentation formalism called assumption-based argumentation (ABA) [4], where, briefly put, arguments are derivations starting off from assumptions via a given set of derivation rules. A (possibly asymmetric) contrary relation decides conflicts between arguments. Altogether arguments and directed conflicts (attacks) are referred to as argumentation frameworks (AFs) [13].

Example 1. *Consider an ABA framework consisting of five assumptions* $\{a, b, c, d, e\}$ *and three rules:* $(p \leftarrow e)$*,* $(\overline{e} \leftarrow c, d)$*, and* $(\overline{d} \leftarrow c)$*. That is, from assumption* e *we can derive* p*, from assumptions* c *and* d *we can derive* $\overline{e}$*, and from assumption* c *we can derive* $\overline{d}$*. As for (asymmetric) contraries, let* $\overline{d}$ *be the contrary of* d*,* $\overline{e}$ *be the contrary of* e*,* b *be the contrary of* c*, and finally* a *the contrary of* b*. These ingredients lead to eight different arguments that can be instantiated from this ABA framework. All of these directly correspond to derivations based on the assumptions via the rules. Figure 1 shows all eight arguments and their directed conflicts. For instance, argument* A_3 *attacks* A_4 *because the former concludes* $\overline{d}$*, the contrary of* d*, which is an assumption in* A_4*.*

Let us consider ways of argumentative acceptance of atom p *and the prominent approach of admissibility and credulous acceptance. Reasoning on ABA frameworks can be defined via arguments: finding a set of non-conflicting arguments that defends itself* *against counterarguments and concludes p. It holds that* $\{A_1,A_3,A_5\}$ *constitutes an admissible set that concludes p, e.g., the attack from* A_2 *onto* A_3 *is countered by* A_1.

Following recent work [8,9,10], one can look at so-called strongly accepting subframeworks that show parts that are sufficient for acceptance. This notion is defined on the level of arguments. For instance here $\{A_3,A_5\}$ *is a strongly accepting subframework for p. Let us look at the subframework consisting only of these two arguments: there are no conflicts and p can be concluded. "Re-adding"* A_4 *leads to a conflict with* A_5 *(the argument we need to defend), but it holds that* A_3 *defends* A_5 *against* A_4. *Adding* A_2 *leads to the case that* A_2 *defends* A_5 *against* A_4 *(again making up an admissible set in the subframework concluding p). Adding further arguments again leads to the overall picture: if one commits to* A_3 *and* A_5 *we can safely find admissible sets concluding p in the subframeworks in-between the one with only these two arguments and all arguments.*

On the other hand, when looking at the structured ingredients needed to conclude p, we find that assumption e and two rules $(\overline{d} \leftarrow c)$ *and* $(p \leftarrow e)$ *are sufficient: with these components we can only instantiate argument* A_5 *(for all others some components are missing), and adding any further rules or assumptions leads to cases where we still find an admissible set concluding p. For instance, re-adding components to instantiate* A_4 *requires assumption c, with which, together with* $(\overline{d} \leftarrow c)$, *leads to the case that* A_3 *can be instantiated, again leading to the case that* A_3 *defends* A_5 *against* A_4. *However, here we see a mismatch: considering all arguments that can be instantiated from the strongly accepting assumption e and the two rules leads to the set* $\{A_5\}$ *of arguments. This set does not constitute a strongly accepting subframework when looking only at the argument-level, since we can add* A_4 *(and no other argument), which defeats* A_5.

We address strongly accepting subframeworks in terms of parts of knowledge bases and AFs, and provide the following main contributions.

- We introduce strongly accepting sub-bases both for a general notion of structured knowledge bases, and for the concrete example of ABA.
- We show that strongly accepting subframeworks of a corresponding AF induce a strongly accepting sub-base on the side of the knowledge base, for a generic structured argumentation approach.
- At the same time, as exemplified above, we show that the converse does not hold in general, and point out that by considering "closed" AFs (containing all arguments and attacks from base), and an adaption of strong acceptance on AFs addresses this issue.
- We show that under mild assumptions, a strongly accepting subframework can be bound polynomially on the argument-level in terms of the original knowledge base. This indicates that even though AFs corresponding to a knowledge base might be large (exponentially-sized), a "witness" for acceptability can be bound polynomially in size.

2. Preliminaries

Abstract Argumentation. We fix a non-finite background set $\mathscr{U}$. An argumentation framework (AF) [13] is a directed graph $F = (A,R)$ where $A \subseteq \mathscr{U}$ represents a set of arguments and $R \subseteq A \times A$ models *attacks* between them. Let $\mathbb{F}$ be the set of all AFs over

$\mathscr{U}$. For two arguments $x,y \in A$, if $(x,y) \in R$ we say that x *attacks* y as well as x *attacks* (the set) E given that $y \in E \subseteq A$. A set $E \subseteq A$ attacks $a \in A$ if $\exists b \in E$ with $(b,a) \in R$. We let $E_F^+ = \{x \in A \mid E \text{ attacks } x\}$ for a set $E \subseteq A$. For two AFs $F = (A,R)$ and $G = (B,S)$ we define the $\subseteq$ relation component-wise, i.e. $F \subseteq G$ if $A \subseteq B$ and $R \subseteq S$.

A set $E \subseteq A$ is *conflict-free* in F iff for no $x,y \in E$, $(x,y) \in R$. We say E *defends* an argument x if E attacks each attacker of x. A conflict-free set E is *admissible* in F ($E \in ad(F)$) iff it defends all its elements. Given an AF $F = (A,R)$ a *semantics* σ returns a set of subsets of A. These subsets are called σ*-extensions*. In this paper we consider so-called *complete*, *grounded*, *preferred*, and *stable* semantics (abbr. *co*, *gr*, *pr*, *stb*).

Definition 1. *Let $F = (A,R)$ be an AF and $E \in ad(F)$.*

- *$E \in co(F)$ iff E contains all arguments it defends;*
- *$E \in gr(F)$ iff E is $\subseteq$-minimal in $co(F)$;*
- *$E \in pr(F)$ iff E is $\subseteq$-maximal in $co(F)$;*
- *$E \in stb(F)$ iff $E^+ = A \setminus E$.*

Assumption-based Argumentation. We assume a deductive system $(\mathscr{L},\mathscr{R})$, where $\mathscr{L}$ is a formal language, i.e. a set of sentences, and $\mathscr{R}$ is a set of inference rules over $\mathscr{L}$. A rule $r \in \mathscr{R}$ has the form $a_0 \leftarrow a_1,\ldots,a_n$ with $a_i \in \mathscr{L}$. We denote the head of r by $head(r) = a_0$ and the (possibly empty) body of r with $body(r) = \{a_1,\ldots,a_n\}$.

Definition 2. *An ABA framework is a tuple $(\mathscr{L},\mathscr{R},\mathscr{A},\overline{})$, where $(\mathscr{L},\mathscr{R})$ is a deductive system, $\mathscr{A} \subseteq \mathscr{L}$ a non-empty set of assumptions, and a contrary function $\overline{} : \mathscr{A} \rightarrow \mathscr{L}$.*

Assumption 1. *In this work, we focus on ABA frameworks which are* flat*, i.e., for each rule $r \in \mathscr{R}$, $head(r) \notin \mathscr{A}$ (no assumption can be derived), and* finite*, i.e., $\mathscr{L}$, $\mathscr{R}$, $\mathscr{A}$ are finite; moreover, each rule is stated explicitly (given as input).*

Given an ABA framework $D = (\mathscr{L},\mathscr{R},\mathscr{A},\overline{})$, a tree-based argument is a finite labeled rooted tree t, also denoted by $A \vdash_{R'} p$ with $A \subseteq \mathscr{A}$, $R' \subseteq \mathscr{R}$, and $p \in \mathscr{L}$, s.t. the root is labeled with p. Moreover, each leaf is labeled by an assumption $a \in \mathscr{A}$ or a dedicated symbol $\top \notin \mathscr{L}$. The set of all labels of leaves is A, and each internal node is labeled with the head of a rule $r \in R'$ s.t. the set of labels of children of this node is equal to the body of r or $\top$ if the body is empty. For each $r \in R'$ there must be such a corresponding internal node. We write $A \vdash p$ if there is some $R' \subseteq \mathscr{R}$ s.t. $A \vdash_{R'} p$. An ABA framework induces an AF as follows.

Definition 3. *The associated AF $F_D = (A,R)$ of an ABA $D = (\mathscr{L},\mathscr{R},\mathscr{A},\overline{})$ is given by $A = \{S \vdash p \mid \exists R' \subseteq \mathscr{R} : S \vdash_{R'} p\}$ and attack relation $(S \vdash p, S' \vdash p') \in R$ iff $p \in \overline{S'}$.*

Semantics of ABA frameworks can then be direct taken as σ-extensions of the associated AFs.

Notion of Credulous Acceptance As for acceptance, we consider credulous acceptance. Given an ABA framework and a semantics σ, an atom $p \in \mathscr{L}$ is (credulously) accepted under σ if there is a σ-extension on the associated AF with an argument $A \vdash p$ in the extension. When looking only at the argument-level, an atom is, likewise, (credulously) accepted if there is a σ-extension concluding the atom. Formally, we let $crd_\sigma(F) = \bigcup_{E \in \sigma(F)} conc(E)$ for a semantics σ and AF F, and with $conc(E) = \{p \mid (A \vdash p) \in E\}$ (i.e., collecting all conclusions of arguments). For an ABA framework D and its associated AF F_D, we let $crd_\sigma(D) = crd_\sigma(F_D)$, i.e. the semantics of ABA frameworks is defined via AF instantiation.

3. A General View on Structured Argumentation

Before delving into defining the notion of strongly accepting subframeworks on ABA, we first define a general view on structured argumentation in order to broaden our scope. We consider a general approach to structured argumentation formalisms in line with ABA. For our purposes, three ingredients are important:

- a definition of structured knowledge bases,
- a translation to AFs,
- a function extracting components of the knowledge base from an instantiated AF.

Formally, a knowledge base is an (abstract) structure B which we simply define as a set for ease of presentation. That is, B can be seen as a set composed of ingredients making up the knowledge base. By definition of sets, we arrive at sub-bases by referring to the $\subseteq$ relation. For instance, $\emptyset$ is the empty knowledge base. Given a knowledge base, we need a function that instantiates the knowledge base as an AF. We denote this function as af. We also consider a function extracting a knowledge base (back) from an AF, denoted by kb. These mappings af and kb were used in a similar fashion before [12], but not presented in the same depth and not connected to strong acceptance.

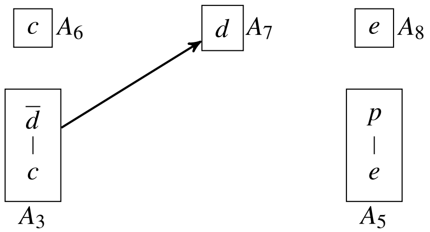
Definition 4. *A knowledge base is a set B. Define the mapping* $af : 2^B \to \mathbb{F}$. *We define* $\mathbb{F}_B = \{F \mid F \subseteq af(B)\}$ *as the set of sub-frameworks of the AF instantiated from B.*

In order to apply our general definitions to ABA frameworks D, we identify $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\ })$ with the set $\mathcal{R} \cup \mathcal{A}$ of rules and assumptions. Recall that each ABA framework D induces an associated AF F_D as defined above. We can thus apply our general proposal to ABA in a naturally way by letting $af_{ABA}(D) := F_D$. Given a fixed ABA framework D, the mapping $af_{ABA} : 2^D \to \mathbb{F}$ formalizes all conceivable possibilities to instantiate D using only a subset of the rules and assumptions. Thereby, we will sometimes work with a technically different ABA framework containing fewer assumptions. With a little notational abuse, we always assume that the contrary function $\bar{\ }$ is suitably restricted to the considered set of assumptions.

Example 2. *In this example, we extend the ABA framework from our motivating example* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{\ })$, *i.e. we have* $\mathcal{A} = \{a, b, c, d, e\}$, $\bar{c} = b$ *as well as* $\bar{a} = b$ *and rules*

$$p \leftarrow e. \qquad \bar{e} \leftarrow c, d. \qquad \bar{d} \leftarrow c. \qquad p \leftarrow r.$$

where "$p \leftarrow r.$*" is a novel rule which is not applicable. For brevity, we use* $\bar{d}$ *and* $\bar{e}$ *to mean fresh symbols without explicating them. The mapping* af_{ABA} *applied to D returns the argumentation framework depicted in Figure 1. If we restrict* $\mathcal{R}$ *to the set of rules* $\{(p \leftarrow e.), (\bar{d} \leftarrow c.)\}$ *and* $\mathcal{A}$ *to* $\{c, d, e\}$, *then the corresponding AF is the sub-graph consisting of* A_3, A_5, A_6, A_7, *and* A_8*:*



For the usual instantiation procedures known from the literature, not every AF $F \in \mathbb{F}_B$ corresponds directly to some subset of the knowledge base. Consider e.g. our running example. There is no subset of $\mathscr{R} \cup \mathscr{A}$ resulting in the AF containing the argument A_5 only since constructing A_5 requires assumption e which would induce A_8 as well.

Moving from a knowledge base to an induced AF is a standard procedure in structured argumentation formalisms. For our investigation, we need to connect AFs and structured bases in both directions. Therefore, we require a formal tool to extract (parts of) the given knowledge base from (parts of) the instantiated AF.

Definition 5. *Let B be a knowledge base. Define the mapping $kb : \mathbb{F}_B \to 2^B$.*

That is, for a knowledge base B, each sub-framework $F \in \mathbb{F}_B$ is mapped to some subset $kb(F) = B' \subseteq B$ of the original knowledge base. Intuitively, one may e.g. think of those parts of the knowledge base which are necessary to construct the arguments occurring in F.

As we already mentioned, not every AF $F \in \mathbb{F}_B$ is induced by some $B' \subseteq B$. Therefore, it holds that *af* and *kb* are in general not inverse to each other. However, for some of our technical results we require the two mappings to correspond to each other in a certain sense. As already mentioned, the intuitive idea is that when considering a sub-framework $F \in \mathbb{F}$, in $kb(F)$ we collect all components of the knowledge base which are necessary to construct F. If we then apply *af* again obtaining $af(kb(F))$, we expect F (and potentially further arguments) to be constructible again, for otherwise our selected components in $kb(F)$ would not suffice for our intended purpose.

Definition 6. *Let B be a knowledge base. We call af and kb* well-behaved *if for all $F \in \mathbb{F}_B$ it holds that $F \subseteq af(kb(F))$.*

An illustration of this notion is depicted in Figure 2. Inspecting the relationship $F \subseteq af(kb(F))$ reveals that if this inclusion is proper, i.e., $F \subsetneq af(kb(F))$, we find a kind of (apparent) "closure" operator, i.e. the composition of *af* and *kb*. Intuitively, if $af(kb(F))$ contains more arguments than F itself, then further arguments can be constructed without the necessity to make use of additional components of the knowledge base.

Definition 7. *Let B be a knowledge base. We call an AF F* closed *if there is some $B' \subseteq B$ s.t. $F = af(B')$. We call af and kb* strictly well-behaved *if they are well behaved and in addition $F = af(kb(F))$ holds for all closed AFs F.*

Let us now demonstrate how we can define a natural mapping *kb* in the context of ABA. The attentive reader may realize that for *kb* to be reasonably defined we need to be able to extract the knowledge base from the given instantiated AF. This is possible

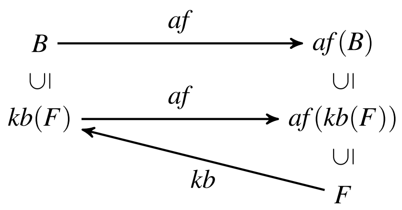


Figure 2. For a given knowledge base B, function $af(B)$ results in the associated AF of B. For an AF F that is a sub AF of $af(B)$, inspecting its components ($kb(F)$) leads to a sub part of B (potentially proper). Applying af on the sub part may lead to a potential super framework of F (namely $af(kb(F))$). The composite function $af(kb(\cdot))$ can be interpreted as a closure operation.

for ABA as long as we can be sure the knowledge base D does not contain any hidden information which is not reflected in F_D.

The following notion suffices to ensure that all information included in the ABA framework are made explicit in the selection of all arguments. It simply states that for each atom $p \in \mathscr{L}$, there is at least one tree-based argument inferring it.

Definition 8. *We call an ABA framework $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, ^-)$* trim *if $\mathscr{L} = Th_D(\mathscr{A})$.*

Interestingly, assuming that D is trim already suffices to rebuild the whole ABA framework by inspecting the constructed arguments. From a technical point of view, we want to emphasize however that in our definition of an argument, the whole tree is stored.

Proposition 1. *Let $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, ^-)$ be a trim ABA framework and let $F_D = (A, R)$ be the associated AF. Then*

- *$\mathscr{L}$ is the union of all labels of roots occurring in A,*
- *$\mathscr{A}$ is the union of all labels of leaves except $\top$ occurring in A,*
- *$a_0 \leftarrow a_1, \dots, a_n \in \mathscr{R}$ iff there is some $t \in A$ s.t. a_0 is a label of a node in t and its children are labeled $a_1, \dots, a_n$.*

Proof. ($\mathscr{L}$) We have $\bigcup_{t \in A} root(t) = Th_D(\mathscr{A}) = \mathscr{L}$ where the first "=" holds by definition and the second since D is trim.

($\mathscr{A}$) We show $\bigcup_{t \in A} leaves(t) \setminus \{\top\} = \mathscr{A}$. The inclusion $\subseteq$ is clear. For the other direction note that each assumption $a \in \mathscr{A}$ induces a tree-based argument $\{a\} \vdash a$.

($\mathscr{R}$) The ($\Leftarrow$)-direction follow from the way argument trees are constructed. Regarding ($\Rightarrow$) let $r = a_0 \leftarrow a_1, \dots, a_n \in \mathscr{R}$. Since D is trim, there are tree-based arguments $t_1, \dots, t_n \in A$ with $root(t_i) = a_i$ for $1 \leq i \leq n$. Therefore, there is a tree-based argument t stemming from $t_1, \dots, t_n$ and the rule r, i.e. $t \in A$ s.t. $root(t) = a_0$ and the children of the root are labeled with $a_1, \dots, a_n$. $\square$

Inspired by Proposition 1, for a given AF F we let $kb_{ABA}(F)$ be the set $\mathscr{R} \cup \mathscr{A}$ of rules and assumptions as described in the proposition, i.e. $a_0 \leftarrow a_1, \dots, a_n \in kb_{ABA}(F)$ iff there is some $t \in A$ s.t. a_0 is a label of a node in t and its children are labeled $a_1, \dots, a_n$; $a \in kb_{ABA}(F)$ iff $a \neq \top$ and there is some leave labelled a. With a little notational abuse we denote the induced ABA framework with $(\mathscr{L}, kb_{ABA}(F_D), ^-)$.

Interestingly, D does not need to be trim in order to find all the necessary components of the knowledge base, since we can simply ignore rules which are not applicable. In the following we establish that af_{ABA} and kb_{ABA} are strictly well-behaved, even without restricting our attention to trim ABA frameworks.

Proposition 2. *Let $D = (\mathscr{L}, \mathscr{R}, \mathscr{A}, ^{-})$ be an ABA framework and let D' be induced by $kb_{ABA}(F)$, i.e. $D' = (\mathscr{L}, kb_{ABA}(F_D), ^{-})$. Then $F_D = F_{D'}$.*

Proof. Set $F_D = (A, R)$ and $F_{D'} = (A', R')$. We have $kb_{ABA}(D) \subseteq \mathscr{R} \cup \mathscr{A}$ (part of the proof of Proposition 1 which does not require D to be trim). Moreover, $kb_{ABA}(D) \cap \mathscr{A} = \mathscr{A}$ is clear since each assumption induces some argument. By definition of the instantiation procedure, $F_{D'} \subseteq F_D$. Since attacks are uniquely determined by the tree-based arguments, it suffices to show that $A \subseteq A'$.

Suppose the contrary, i.e. take $t \in A \setminus A'$. Without loss of generality, assume that each proper sub-argument in t occurs in A'. Since D and D' share the same assumptions, the root label of t is no assumption, say a_0. Let $a_1, \ldots, a_n$ be the label of the children. By definition $a_0 \leftarrow a_1, \ldots, a_n \in kb_{ABA}(D)$ and hence $t \in A'$; contradiction. □

Example 3. *For D with the AF F_D from Example 2, $kb_{ABA}(F_D)$ consists of the rules*

$$p \leftarrow e. \qquad \overline{e} \leftarrow c, d. \qquad \overline{d} \leftarrow c.$$

and assumptions $\mathscr{A} = \{a, b, c, d, e\}$ which can be extracted from the tree-based instantiated arguments occurring in the AF (see Figure 1); the dummy rule "$p \leftarrow r.$" is lost. Nonetheless, $af_{ABA}(kb_{ABA}(F_D)) = F_D$.

Consider an AF F' consisting only of argument A_5 (with assumption e and rule $p \leftarrow e$). This AF is not closed. It holds that $F' \subsetneq af_{ABA}(kb_{ABA}(F')) = F''$ with F'' having the two arguments A_5 and A_8. Confirming our intuition, F'' is now closed.

Corollary 1. *The mappings af_{ABA} and kb_{ABA} are strictly well-behaved.*

We now have settled the syntax of our approach. Regarding the semantics, let us consider a generic mapping *acc* that returns for both knowledge bases and AFs a set of "accepted" atoms.

Definition 9. *The mappings acc and af are called* compatible *if it holds that $a \in acc(B)$ iff $a \in acc(af(B))$ for each knowledge base B.*

Intuitively, compatibility simply states that when instantiating a knowledge base, the corresponding semantics are preserved. It is well-known that this is the case for ABA.

Proposition 3. *For the ABA and AF semantics we consider, crd_σ and af_{ABA} are compatible (for fixed $\mathscr{L}$, and $^{-}$).*

We remark that our general view and the condition of *af* and *kb* being well-behaved ($F \subseteq af(kb(F))$) have a certain relation to Galois connections. Highlighting only the essential, if $kb(F) \subseteq B$ implies $F \subseteq af(B)$ (*), one can show that $F \subseteq af(kb(F))$ follows. Condition (*) can be seen as "one direction" of the requirement for Galois connections. Intuitively, condition (*) appears plausible for structured argumentation: if B contains all ingredients to instantiate F, then $af(B)$ should contain all of F, as well.

Proposition 4. *If $kb(F) \subseteq B$ implies $F \subseteq af(B)$, it holds that $F \subseteq af(kb(F))$.*

Proof. By definition, $kb(F) \subseteq kb(F)$ holds and $F \subseteq af(kb(F))$ holds by assumption. □

4. Strongly Accepting Subframeworks

Next we discuss our notion of strong acceptance, utilizing earlier works [8,10].

4.1. Basic Definitions

The idea behind a strongly accepting sub-framework is that one aims to find a sub-framework $B' \subseteq B$ accepting a certain atom (argument), but accounts for non-monotonicity by requiring that this property survives moving to supersets of B' within B. We first define this idea on knowledge bases.

Definition 10. *Let B a knowledge base. A set $B' \subseteq B$* strongly accepts *a if $a \in acc(B'')$ for all B'' such that $B' \subseteq B'' \subseteq B$.*

As mentioned earlier, strong acceptance can be defined on AFs [8,10].

Definition 11. *Let B a knowledge base and $F = af(B)$. A sub-AF $F' \in \mathbb{F}_B$* strongly accepts *a if $a \in acc(F'')$ for all F'' such that $F' \subseteq F'' \subseteq F$.*

Example 4. *Consider our running Example 2. As we already discussed in the introduction, the sub-AF consisting of $\{A_3, A_5\}$ strongly accepts A_5 (and hence the atom p). From the point of view of the knowledge base, we require the assumptions $\mathscr{A}' = \{c, e\}$ and rules $\mathscr{R}' = \{(\bar{d} \leftarrow c, e.), (p \leftarrow e.)\}$ to construct these arguments. The reader may verify that indeed, $\mathscr{A}' \cup \mathscr{R}'$ strongly accepts p (note that this is a superset to the one discussed in the introduction; we come back to a smaller one below).*

The observation we made in the previous example is no coincidence: given a strongly accepting sub-graph of $af_{ABA}(B)$, under mild conditions we can find a strongly accepting subset of B by applying the mapping kb_{ABA}. The following proposition formalizes this result.

Proposition 5. *Let B be a knowledge base, and $F' \in \mathbb{F}_B$. Suppose af is $\subseteq$-monotone, and af, kb, and acc are well-behaved and compatible. If F' strongly accepts a, then $kb(F')$ strongly accepts a.*

Proof. Assume that F' strongly accepts a and $kb(F') = B'$ does not. Then there is a B'' with $B' \subseteq B'' \subseteq B$ with $a \notin acc(B'')$. It holds that $af(B') \subseteq af(B'')$ (by monotonicity of af) and $F' \subseteq af(kb(F'))$ (by af and kb being well-behaved). Then $F' \subseteq af(kb(F')) = af(B') \subseteq af(B'')$. Moreover, since $af(B)$ is $\subseteq$-maximal, it follows that $F' \subseteq af(B'') \subseteq af(B)$. By assumption that F' is strongly accepting a, it follows that $a \in acc(af(B''))$, which implies $a \in acc(B'')$, a contradiction. □

The opposite direction does not hold in general, also not for concrete instantiations (e.g., ABA and ASPIC$^+$ [3]) as discussed partially before and more concretely in the following counter-example.

Example 5. *From our running example ABA D consider the rules $\mathscr{R}'$:*

$$\bar{d} \leftarrow c. \qquad\qquad p \leftarrow e.$$

and $\mathscr{A}' = \{e\}$. The set $D' = \mathscr{A}' \cup \mathscr{R}'$ strongly accepts p in D which can be seen as follows. In order to prevent p from being acceptable, the argument A_4 is required. However, for this we would need to add assumptions c and d as well which in turn allows to construct A_3. We already know however that presence of A_3 and A_5 suffice to strongly accept p. On the other hand, the AF $af_{ABA}(D')$ consists of the argument A_5 from Figure 1 only which does not suffice to strongly accept p: we can simply add A_4.

However, if *af* is a bijection and *kb* its inverse, then the converse is also true.

Proposition 6. *Let B a knowledge base, and $F' \in \mathbb{F}_B$ as well as $B' \subseteq B$. Suppose af and kb are $\subseteq$-monotone, and af, kb, and acc are compatible s.t. $af : 2^B \to \mathbb{F}_B$ is a bijection and $kb = af^{-1}$. If F' strongly accepts a, then $kb(F')$ strongly accepts a. If B' strongly accepts a, then $af(B')$ strongly accepts a.*

To summarize, the notion of strong acceptance can be naturally defined for both the AFs as well as the underlying knowledge base. Since not every AF $F \in \mathbb{F}_B$ is induced by some subset of the knowledge base B, it is in general not true that strong acceptance is preserved when applying *af* resp. *kb*. However, as formalized by Proposition 5, under mild conditions it can be translated back from the AF to the knowledge base. As our results regarding ABA demonstrate, the preconditions for Proposition 5 hold for ABA.

If one restricts strong acceptance on the AF side to only closed AFs in $\mathbb{F}_B$, in addition to having *af* being a bijective with *kb* its inverse, we can apply Proposition 6 to conclude that strong acceptance transfers in both ways.

4.2. Strong Acceptance and Size of Subframeworks

In the general case, an associated AF may not be small w.r.t. the structured knowledge base it was instantiated from. For instance, an AF associated to an ABA framework may be exponential in size [14] (the example given there applies to ABA as well). Nevertheless, as we show, in many cases one can bound strongly accepting subframeworks *both* on the knowledge base and argument level.

To formalize this idea, we will make use of claim-augmented AFs (CAFs) [15]. A CAF is a triple $\mathscr{F} = (A, R, cl)$ where $F = (A, R)$ is an AF and $cl : A \to \mathscr{C}$ assigns a claim $c \in \mathscr{C}$ to each argument in A; $\mathscr{C}$ is a countably-infinite set. We let $cl(E) = \{cl(e) \mid e \in E\}$ for a set E of arguments. In the literature, semantics for CAFs have been introduced and formally investigated, but we are only interested in the claims as additional information. We thus let $\sigma(\mathscr{F}) = \sigma(F)$ for each semantics σ considered in this paper. We assume that our mappings *af*, *kb*, and *acc* naturally extend to CAFs.

Example 6. *Our running example ABA frameworks yields a CAF $\mathscr{F} = (A, R, cl)$ where the underlying $F = (A, R)$ corresponds to the AF from Figure 1. Claims of arguments correspond to their conclusions, i.e. we let $cl(A_3) = \bar{d}$ and analogously for the other A_i.*

When constructing arguments corresponding to an ABA framework D, out-going attacks are naturally characterized by the conclusions of an argument. Viewing the AF as a CAF by defining *cl* as shown above, this procedure yields a well-formed CAF. In the following, we assume that the CAFs we work with possess this feature.

Assumption 2. *For a given KB B, $af(B) = \mathscr{F} = (A, R, cl)$ is well-formed in the sense that $cl(a) = cl(b)$ implies $a^+ = b^+$ for each argument $a, b \in A$.*

We remark that this assumption holds for many structured argumentation approaches, such as ABA, but not all: in case of preferential approaches (e.g., ASPIC$^+$ [3]) one can find counterexamples. We make use of one additional technical assumption that holds for several structured argumentation formalisms.

Assumption 3. *For a given KB B, it holds that the set of claims in* $af(B)$ *is bounded polynomially by* $|B|$.

This assumption does not hold, e.g., if there is an underlying logic or deductive system of a knowledge base which gives rise to claims not present in original knowledge base (e.g., if a leads to $a \vee b$ leads to $a \vee b \vee c$, ...). Observe however that for our running example of non-preferential ABA instantiations, we have $|cl(A)| \leq |\mathscr{L}|$ and for each framework D, F_D is well-formed.

In the following, we will show that any strongly accepting sub-framework can be reduced to at most $|C|$ arguments, where C is the set of claims occurring in F. The crucial observation is formalized in the following theorem. It states an admissible extension $E \in ad(F)$ with more than $|C|$ claims can be reduced in size. The proof proceeds by removing arguments which do not contribute any novel claim.

Theorem 1. *Let* $\mathscr{F} = (A, R, cl)$ *be well-formed,* $a \in A$*, and the set of claims* $C = cl(A)$ *of* $\mathscr{F}$ *finite. If a set of arguments* E *is admissible in* F *and contains* a*, then there exists an admissible set* E' *with* $a \in E'$ *and* $|E'| \leq |C|$.

Proof. Let E be admissible in F and $a \in E$. If $|E| \not\leq |C|$ then there is a claim $\alpha \in C$ such that more than one argument in E has claim α. Pick any $a \in E$ with $claim(a) = \alpha$. We claim that $E' = E \setminus \{x \in E \mid claim(x) = \alpha, x \neq a\}$ is admissible in F. It holds that E' is conflict-free (since conflict-freeness holds for any subset of a conflict-free set). Suppose there is a $b \in E'$ such that there is a $c \in A$ with $(c, b) \in R$ and there is no $d \in E'$ with $(d, c) \in R$. Since $E' \subseteq E$ it holds that $b \in E$. Since E is admissible there is a $d' \in E$ such that $(d, c) \in R$. Since $d' \in E \setminus E'$, by construction of E', it holds that $claim(d') = \alpha = claim(a)$. By well-formedness, it holds that $(a, c) \in R$, contradicting the assumption that b is not defended by E'. We conclude that E' is admissible in F. Define $C' \subseteq C$ as the set of claims of arguments in E. Finally, pick for each claim $\alpha \in C'$ an argument $x_\alpha \in E$ with the exception that a_α is chosen for the claim of a, and set $E^* = \{x_\alpha \mid x_\alpha \in E, \alpha \in C'\}$. By the statements above it holds that E^* is admissible and contains a. By construction, there is exactly one argument per claim in C'. □

For the other semantics $\sigma \in \{co, gr, pr, stb\}$ we might move to a superset in order to fulfill the semantic-specific requirements, but we can be sure that an admissible set of small size can be extended in a suitable way.

Corollary 2. *Let* $\mathscr{F} = (A, R, cl)$ *be well-formed,* $a \in A$*, and the set of claims* $C = cl(A)$ *of* F *finite. Let* $\sigma \in \{co, gr, pr, stb\}$*. If* $E \in \sigma(F)$*, then there is some* $E_0 \in ad(F)$ *with* $|E_0| \leq |C|$ *and* $E_0 \subseteq E$.

Since extensions induce strongly accepting subsets as formalized in [8] we can now infer that the size of such subsets can be trimmed down to size of at most $|C|$ arguments.

Corollary 3. *Let* $\mathscr{F} = (A, R, cl)$ *be well-formed,* $a \in A$*, and the set of claims* $C = cl(A)$ *of* F *finite. Let* $\sigma \in \{co, gr, pr, stb\}$*. If there is a strongly accepting sub-framework for a, then here is also a strongly accepting sub-framework containing at most* $|C|$ *arguments.*

5. Conclusions

In this paper we revisited strongly accepting subframeworks [10,8,9] (see also [16]), and investigated their connection to structured argumentation frameworks. Based on a generic notion of such structured frameworks, we showed that strongly accepting subframeworks are applicable and generalizable from abstract AFs to structured frameworks, with the concrete formalism ABA being presented here. There is an apparent mismatch between notions on the abstract and structured level, but which can be addressed with careful consideration of, e.g., closed AFs. Moreover, we considered properties connecting to strongly accepting frameworks, in particular we showed that such strongly accepting subframeworks can be bounded polynomially, even if a "full" AF is not bounded polynomially, indicating that notions supporting explanations, based on strongly accepting subframeworks, exhibit interesting size bounds, and open up further investigation.

The most apparent future work directions include the investigation of other concrete structured argumentation formalisms and finding natural and mild conditions ensuring the converse of Proposition 5. Moreover, studying relations to other forms explainability is an intriguing avenue of future research.

References

[1] Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018.

[2] Atkinson K, Baroni P, Giacomin M, Hunter A, Prakken H, Reed C, et al. Towards Artificial Argumentation. AI Magazine. 2017;38(3):25-36.

[3] Modgil S, Prakken H. A general account of argumentation with preferences. Artificial Intelligence. 2013;195:361-97.

[4] Bondarenko A, Dung PM, Kowalski RA, Toni F. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. Artificial Intelligence. 1997;93:63-101.

[5] García AJ, Simari GR. Defeasible Logic Programming: An Argumentative Approach. Theory and Practice of Logic Programming. 2004;4(1-2):95-138.

[6] Besnard P, Hunter A. Elements of Argumentation. MIT Press; 2008.

[7] Cyras K, Rago A, Albini E, Baroni P, Toni F. Argumentative XAI: A Survey. In: Proc. IJCAI. ijcai.org; 2021. p. 4392-9.

[8] Ulbricht M, Wallner JP. Strong Explanations in Abstract Argumentation. In: Proc. AAAI. AAAI Press; 2021. p. 6496-504.

[9] Niskanen A, Järvisalo M. Smallest Explanations and Diagnoses of Rejection in Abstract Argumentation. In: Proc. KR; 2020. p. 667-71.

[10] Saribatur ZG, Wallner JP, Woltran S. Explaining Non-Acceptability in Abstract Argumentation. In: Proc. ECAI. vol. 325 of Frontiers in Artificial Intelligence and Applications; 2020. p. 881-8.

[11] Prakken H, Winter MD. Abstraction in Argumentation: Necessary but Dangerous. In: Proc. COMMA. vol. 305 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2018. p. 85-96.

[12] Wallner JP. Structural constraints for dynamic operators in abstract argumentation. Argument & Computation. 2020;11(1-2):151-90.

[13] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321-57.

[14] Strass H, Wyner A, Diller M. *EMIL*: Extracting Meaning from Inconsistent Language: Towards argumentation using a controlled natural language interface. International Journal of Approximate Reasoning. 2019;112:55-84.

[15] Dvořák W, Woltran S. Complexity of abstract argumentation under a claim-centric view. Artificial Intelligence. 2020;285:103290.

[16] Brewka G, Thimm M, Ulbricht M. Strong inconsistency. Artificial Intelligence. 2019;267:78-117.