Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Splitting Assumption-Based Argumentation Frameworks

Giovanni Buraglio, Wolfgang Dvořák, Stefan Woltran
Institute of Logic and Computation, TU Wien, Austria
{giovanni.buraglio, wolfgang.dvorak, stefan.woltran}@tuwien.ac.at,

Abstract

Assumption-Based Argumentation (ABA) is a well-established formalism for modelling and reasoning over debates, with a wide range of applications. However, the high computational complexity of core reasoning tasks in ABA poses a significant challenge for its applicability. This issue is further aggravated when ABA frameworks (ABAFs) are instantiated into graph-based argumentation formalisms, such as Dung's Argumentation Frameworks (AFs) and Argumentation Frameworks with Collective Attacks (SETAFs). In knowledge representation and reasoning, a key strategy to address computational intractability is to optimise reasoning over a given knowledge base through divide-and-conquer algorithms. A paradigmatic example of this approach is splitting, where extensions of a given framework are computed incrementally, by restricting the search space to sub-frameworks only, and then combining the obtained results. This approach has been successfully applied to AFs, for which also a parametrised version has been introduced under stable semantics. However, the exponential growth produced by the instantiation might undermine the usefulness of splitting on the argument graphs induced by ABAFs. To address this issue, our work investigates the concept of splitting on the knowledge base rather than on its graph-based instantiation. Furthermore, we generalise splitting to its parametrised version for ABAFs.

1 Introduction

Computational models of argumentation in AI (Gabbay et al. 2021) offer formal approaches to represent and reason over debates involving conflicting or uncertain information (Carrera and Iglesias 2015; Dimopoulos, Mailly, and Moraitis 2019; Fan and Toni 2012; Gao et al. 2016; Hadoux, Hunter, and Polberg 2023; Toni 2013). Assumption-Based Argumentation (ABA) (Bondarenko et al. 1997) captures argumentative scenarios by means of so-called ABA frameworks (ABAFs), or ABA knowledge bases, consisting of a set of defeasible sentences (assumptions) and inference rules. Argumentative reasoning can be performed in ABA following two different approaches. The *direct* approach, typically employed by ABA solvers (Lehtonen, Wallner, and Järvisalo 2021a; Lehtonen, Wallner, and Järvisalo 2021b), allows to reason over the ABA knowledge base itself through semantics defined at the level of assumptions. In contrast, the *indirect* approach (Lehtonen et al. 2024) realises reasoning in a two-step process: first an argument graph comprising arguments and their relations is generated from the ABAF, by means of the so-called *instantiation* procedure; then, *semantics* from abstract argumentation are applied to the obtained graph in order to find acceptable sets of arguments, along with the assumptions supporting them.

Although ABA is a well-established formalism for non-monotonic reasoning, with applications in medical decision-making, explainable AI, and causal discovery (Fan et al. 2014; Fan 2018; Russo, Rapberger, and Toni 2024), the high computational complexity of core reasoning tasks in ABA poses a significant challenge for its deployment in practice (Cyras, Heinrich, and Toni 2021). This issue is further aggravated when ABAFs are instantiated into argument graphs, such as Dung's Argumentation Frameworks (AFs) (Dung 1995) or Argumentation Frameworks with Collective Attacks (SETAFs) (Nielsen and Parsons 2006), as the instantiation can be computationally expensive and may result in exponentially large graphs (Lehtonen et al. 2023).

In the context of non-monotonic reasoning, one prominent strategy to address computational intractability is to optimise reasoning over a given knowledge base through divide-and-conquer algorithms. A paradigmatic example of this approach is *splitting*, originally developed for answer-set programming (Lifschitz and Turner 1994) and later adapted to other nonmonotonic formalisms, e.g. default theories (Turner 1996) and recently Abstract Argumentation (Baumann 2011; Baumann et al. 2012; Linsbichler 2014; Baumann, Brewka, and Wong 2011; Liao 2013; Baroni, Giacomin, and Liao 2014). This approach focuses on incrementally computing the extensions of a given argumentation framework by means of the extension of its sub-frameworks, thereby avoiding to consider the entire solution-space of the original framework.

Despite the successful application in graph-based argumentation (Baumann, Brewka, and Wong 2011), splitting, so far, has been neglected for rule-based argumentation systems like Assumption-Based Argumentation. A straightforward attempt to apply splitting in ABA would be to first instantiate the ABAF to a corresponding Dung's AF and then perform AF splitting. However, the exponential number of auxiliary arguments introduced during instantiation might invalidate the usefulness of splitting. To overcome this limitation, we instead consider the instantiation of ABAFs as

SETAFs, which yield more concise graphical representations (König, Rapberger, and Ulbricht 2022). Thus, we first develop a splitting scheme for SETAFs, enabling incremental computation for the indirect approach of reasoning. This method has the advantage of being easily applicable to any formalism that can be instantiated as a SETAF. However, its drawback is that we have to instantiate the knowledge base, which comes with a computational cost, and we cannot even use the splitting to speed up the instantiation step. To address this issue, we present a splitting schema that operates directly on ABA knowledge bases. To this end, this paper makes the following contributions:

- Towards ABA splitting, we generalise existing notions of splitting from AFs to SETAFs (Section 3) for complete, stable, preferred and grounded semantics.
- We then introduce the notion of ABA splitting in Section 4, along with the syntactic adjustments required to establish a splitting theorem, which we prove under complete, stable, preferred and grounded semantics.
- In Section 5 we extend our results to the more general notion of *parameterised splitting* (Baumann et al. 2012), showing that a splitting theorem holds for ABAFs under stable semantics.
- In Section 6 we address computing splittings for SETAFs and ABAFs, by reducing it to a graph splitting problem and reusing dedicated algorithms for AFs.
- In Section 7 we take a closer look at the relationship between the splitting algorithm for ABAFs and the instantiation procedure into SETAFs.
- Finally, after reviewing related work (Section 8), we conclude with a summary and outline directions for future research (Section 9).

The proofs of the remaining results and additional material are available at (Buraglio, Dvorak, and Woltran 2026).

2 Preliminaries

Assumption-Based Argumentation We recall here the basic concepts of assumption-based argumentation (ABA) (Cyras et al. 2018). Debates are represented by means of so-called ABA Frameworks (ABAFs), which consist of a deductive system $(\mathcal{L}, \mathcal{R})$, where $\mathcal{L}$ is a set of sentences, and $\mathcal{R}$ is a set of rules over $\mathcal{L}$. A rule $r \in \mathcal{R}$ has the form $a_0 \leftarrow a_1, \ldots, a_n$ with $a_i \in \mathcal{L}$, $body(r) = \{a_1, \ldots, a_n\}$ and $head(r) = a_0$.

Definition 1. *An ABAF is a tuple $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, where $(\mathcal{L}, \mathcal{R})$ is a deductive system, $\mathcal{A} \subseteq \mathcal{L}$ a set of assumptions, and $^- : \mathcal{A} \to \mathcal{L}$ is a total mapping, called contrary function.*

For a set of assumptions, $S \subseteq \mathcal{A}$ we use $\overline{S}$ to indicate the set of contraries of S. Conversely, we define the partial function $\alpha : \mathcal{L} \to \mathcal{A}$ assigning an assumption to its contrary $b \in \overline{\mathcal{A}}$ such that $\alpha(b) = a$ if $b = \overline{a}$. This generalises to sets of contraries as before. For a set of rules R, we fix $head(R) = \{head(r) \mid r \in R\}$, $body(R) = \{body(r) \mid r \in R\}$. Further, we use $atom(S) = \{p \in \mathcal{L} \mid p \in S \vee \alpha(p) \in S \vee \overline{p} \in S\}$. In what follows, we read $atom(p)$ as $atom(\{p\})$. For a rule $r \in \mathcal{R}$, we say that r is: a fact if $body(r) = \emptyset$; a loop-rule if $\overline{a} = head(r)$ and $a \in body(r)$.

A sentence $q \in \mathcal{L}$ is tree-derivable from $S \subseteq \mathcal{A}$ and rules $R \subseteq \mathcal{R}$, denoted by $S \vdash^R q$, if there is a finite rooted labelled tree T where: the root of T is labelled with q; the set of labels for the leaves of T is equal to S or $S \cup \{\top\}$; and for every inner node v of T there is a rule $r \in R$ such that v is labelled with $head(r)$, and every successor of v is labelled with $a \in body(r)$ or $\top$ if $body(r) = \emptyset$. We sometimes write $S \vdash q$ instead of $S \vdash^R q$ if it does not cause confusion. Moreover, we call $Th_D(S) = \{p \in \mathcal{L} \mid S \vdash p\}$ the theory of S w.r.t. the ABAF D. Throughout the paper, we assume that ABAFs do not contain *dummy rules*, whose body is not derivable from any set of assumptions.

Definition 2. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF. A set $S \subseteq \mathcal{A}$ attacks $T \subseteq \mathcal{A}$ if $S' \vdash \overline{a}$ for some $S' \subseteq S$ and $a \in T$. A set S is conflict-free in an ABAF D ($S \in cf(D)$) if it does not attack itself; S defends T iff it attacks each attacker of T ; S is closed iff $S \vdash a$ implies $a \in S$; S is admissible ($S \in adm(D)$) if it is conflict-free and defends itself.*

We say a set S of assumptions attacks an assumption a if S attacks the singleton $\{a\}$. Moreover, for a set $B \subseteq \mathcal{A}$ we say that S attacks B if S attacks some $b \in B$. We use $S_R^+ = \{a \in \mathcal{A} \mid S \vdash^R \overline{a}\}$ and define the *range* of S w.r.t. $R \subseteq \mathcal{R}$ as $S_R^\oplus = S \cup S_R^+$. In this paper, we assume ABAFs to be flat, unless specified otherwise. We call an ABAF flat if every set S of assumptions is closed, and *non-flat* otherwise. We next recall definitions for grounded, complete, preferred, and stable ABA semantics (abbr. *grd*, *com*, *pref*, *stb*).

Definition 3. *Let D be an ABAF and let $S \in adm(D)$. $S \in com(D)$ iff S contains every assumption set it defends; $S \in grd(D)$ iff S is $\subseteq$-minimal in $com(D)$; $S \in pref(D)$ iff S is $\subseteq$-maximal in $com(D)$; $S \in stb(D)$ iff S attacks each $\{x\} \subseteq \mathcal{A} \setminus S$. We call $\sigma(D)$ the set of σ-extensions of D.*

SETAF Instantiation Reasoning in ABAFs is often performed on graphs induced from the instantiation procedure. Due to their popularity and simple structure, this step has traditionally been performed by means of Dung's AFs (Dung 1995; Bondarenko et al. 1997; Cyras et al. 2018), i.e., directed graphs where nodes and their (binary) relation are interpreted as arguments and attacks among them. However, this representation does not capture the possibility of having multiple assumptions attacking another directly, which might result in a large number of auxiliary arguments. Thus, in recent years, a hyper-graph representation (König, Rapberger, and Ulbricht 2022) of ABAFs, based on argumentation frameworks with collective attacks (SETAFs) (Nielsen and Parsons 2006), has become increasingly popular (Buraglio et al. 2024b; Russo, Rapberger, and Toni 2024; Berthold, Rapberger, and Ulbricht 2024; Dimopoulos et al. 2024).

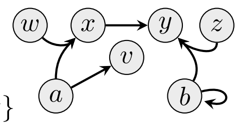
Definition 4. *A SETAF is a pair $SF = (A, R)$ where A is a finite set of* arguments*, and $R \subseteq 2^A \times A$ is the* attack relation*. For an attack $(T, h) \in R$ we call T the* tail *and h the* head *of the attack. We write (t, h) to denote the set-attack $(\{t\}, h)$. For $S \subseteq A$, we say S* attacks *an argument $a \in A$*

if there is an attack $(T, a) \in R$ with $T \subseteq S$. Moreover, for a set $B \subseteq A$ we say that S attacks B if S attacks some $b \in B$. We use $S_R^+ = \{a \in A \mid S \text{ attacks } a\}$ and define the range *of S w.r.t. R as $S_R^\oplus = S \cup S_R^+$. By AFs we refer the class of SETAFs where all attacks $(T, h) \in R$ are such that $|T| = 1$.*

For all semantics under our considerations, an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ can be instantiated as the (equivalent) SETAF $SF_D = (A_D, R_D)$ where $A_D = \mathcal{A}$ and $(S, a) \in R_D$ iff $S \vdash \overline{a}$ (König, Rapberger, and Ulbricht 2022).

Example 1. *For the ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ (left) and its corresponding SETAF SF_D (right), it holds that $\{a, w, z\} \in pref(D) = pref(SF_D)$.*

$$\begin{aligned}
\mathcal{A} = \;& \{a, b, v, w, x, y, z\} \\
\mathcal{L} = \;& \mathcal{A} \cup \overline{\mathcal{A}} \cup \{p\} \\
\mathcal{R} = \;& \{\overline{b} \leftarrow b,\ p \leftarrow a,\ \overline{v} \leftarrow a, \\
& \overline{x} \leftarrow p, w,\ \overline{y} \leftarrow x,\ \overline{y} \leftarrow \overline{b}, z\}
\end{aligned}$$



Notice that such a mapping is many-to-one, i.e., several ABAFs correspond to the same SETAF. This is because the SETAF instantiation only remembers the attacks between assumptions but forgets all non-assumptions. In the above example, we lose the sentence p when instantiating the derivation $\{a, w\} \vdash \overline{x}$ build via $\overline{x} \leftarrow p, w$ and $p \leftarrow a$ into $(\{a, w\}, x)$. On the other hand, SETAFs can be seen — syntactically — as a fragment of flat ABAFs, by modelling each attack (S, a) via a dedicated rule $\overline{a} \leftarrow S$.

Splitting of AFs We now recall the splitting approach for AFs (Baumann 2011). A splitting identifies two sub-frameworks F_1 and F_2 separated by a set of attacks going from F_1 to F_2. Then, the information contained in an extension of F_1 is propagated, computing the so-called *reduct* of F_2 accordingly.

Definition 5. *Let $F = (A, R)$ be an AF, $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$ two sub-frameworks of F s.t. $A_1 \cap A_2 = \emptyset$, $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ with $R_3 \subseteq A_1 \times A_2$. The triple (F_1, F_2, R_3) is called a* splitting *of F. For such a splitting and a set $E \subseteq A_1$, the (E, R_3)*-reduct *is the AF $AF' = (A', R')$ with $A' = A_2 \setminus E_{R_3}^+$ and $R' = R_2 \cap (A' \times A')$. Moreover, the set of* undecided arguments *w.r.t. $E \subseteq A_1$ is $U_E = A_1 \setminus E_{R_1}^\oplus$.*

The reduct is designed to take care of the arguments attacked by the set E. To account for the propagation of undecided arguments w.r.t. E, a further *modification* is needed: self-attacks are propagated from F_1 to arguments in F_2.

Definition 6. *Let (F_1, F_2, R_3) be a splitting for an AF F and E an extension of F_1. Moreover, take $F_2' = (A_2', R_2')$ as the (E, R_3)-reduct of F_2 and U_E as the set of undecided arguments w.r.t. E. The (U_E, R_3)-modification of F_2 is:*

$$mod_{U_E}^{R_3}(F_2') = (A_2', R_2' \cup \{(b, b) \mid \exists a \in U_E : (a, b) \in R_3\}).$$

Using these definitions, Baumann (2011) has shown that it is possible to split the AF and compute the extensions for each sub-framework incrementally such that their combination yields extensions of the original framework.

Theorem 1. *(Baumann 2011) Let (F_1, F_2, R_3) be a splitting for an AF $F = (A, R)$ and $\sigma \in \{adm, stb, com, pref, grd\}$.*

1. *If $E_1 \in \sigma(F_1)$ and $E_2 \in \sigma(mod_{U_E}^{R_3}(F_2'))$, then $E_1 \cup E_2 \in \sigma(F)$.*
2. *If $E \in \sigma(F)$, then $E \cap A_1 \in \sigma(F_1)$ and $E \cap A_2 \in \sigma(mod_{U_E}^{R_3}(F_2'))$.*

Later, this idea has been generalised by relaxing the strict separation requirement, which significantly narrows the applicability of splitting, introducing so-called *parametrised splitting* (Baumann et al. 2012). Instead of demanding that the first part is completely unaffected by the second, it allows some forms of interaction. This generalisation is captured by the notion of *quasi-splitting*, where arguments in F_1 may be externally attacked by arguments in F_2. The goal is to preserve correctness while broadening the applicability of splitting. This is achieved by enriching F_1 with meta-information that encodes facts about potential influences (e.g. attacks) from the second sub-framework. In particular, for each externally attacked argument a, a fresh argument a' is added to F_1 along with a symmetric attack on a, enforcing a choice between a and a' in F_1. Then, F_2 is modified accordingly: the previous choices are propagated in the second sub-framework via the reduct as well as additional nodes and attacks. Stable extensions of the entire AF are then recovered by composing compatible solutions from the two modified sub-frameworks.

3 Splitting Collective Attacks

Towards a splitting approach for ABA we first introduce a splitting scheme for argumentation frameworks with collective attacks (SETAFs). First, due to their correspondence with ABAFs this paves the way for the splitting scheme on the ABA knowledge base that we will introduce in the next second. Second, this provides a divide-and-conquer methodology to enhance existing solvers for SETAFs (Dvořák, Greßler, and Woltran 2018; Greßler, Dvořák, and Woltran 2024), and thus allows for splitting when solving ABAFs via instantiations to SETAFs.

We introduce a notion of splitting for SETAFs that generalises the one for Dung-style AFs.

Definition 7. *Let $SF = (A, R)$ be a SETAF, $SF_1 = (A_1, R_1)$ and $SF_2 = (A_2, R_2)$ two sub-frameworks of SF such that $A_1 \cap A_2 = \emptyset$, $A = A_1 \cup A_2$ and $R = R_1 \cup R_2 \cup R_3$ with $R_3 \subseteq \left((2^{A_1} \setminus \{\emptyset\}) \cup 2^{A_2}\right) \times A_2$. We call a* splitting *of SF the triple (SF_1, SF_2, R_3). Moreover, we call R_3 the set of* links *w.r.t. (SF_1, SF_2, R_3) and say that a link is* undecided *if no argument in its tail is defeated (i.e. attacked by an extension), but at least one is undecided.*

As for AFs, the general idea is to compute extensions of SF as a combination of extensions of SF_1 and SF_2. Due to the links from SF_1 to SF_2 we have to modify SF_2 according to the extension(s) of SF_1 to account for the prior accepted and rejected arguments. Following Baumann (2011), we introduce the notions of *reduct* and *modification*, in application to the second part (that is, SF_2) of the original SETAF. Intuitively, the reduct takes care of the arguments in SF_2 that are already defeated by E_1 by deleting them. Moreover, the reduct removes an attack in R_2 if one of the arguments in the tail or the head is defeated by E_1. The attacks of R_3

are first filtered and the simplified by the reduct in order to fit to the new argument set. An attack is neglected if either a tail or the head argument is defeated by E_1, all the tail arguments are in A_1, or there is a tail arguments in A_1 that is not accepted by E_1. The remaining attacks are simplified by removing the arguments of A_1 from the tail of the attack.

Definition 8 (Reduct)**.** *Let (SF_1, SF_2, R_3) be a splitting for a SETAF SF. We define the* (E_1, R_3)-reduct *(or simply reduct) of SF_2 for some extension E_1 of SF_1 as the SETAF $SF_2' = (A_2', R_2')$ with:*

$$\begin{aligned} A_2' =& \{a \in A_2 \mid a \notin (E_1)^+_{R_3}\}; \\ R_2' =& \{(T, h) \in R_2 \mid T \subseteq A_2', h \in A_2'\} \cup \\ & \{(T \setminus A_1, h) \mid (T, h) \in R_3,\ T \setminus A_1 \neq \emptyset, \\ & \qquad T \cap A_1 \subseteq E_1, T \cap (E_1)^+_{R_3} = \emptyset,\ h \in A_2'\}. \end{aligned}$$

When dealing with undecidedness, what guides our intuition towards a certain modification is not the status of the arguments in SF_1, but rather the status of the *links*. Hence, we slightly tweak the original definition and base our notion solely on the undecided *links*.

Definition 9 (Undecided Links)**.** *Given a splitting (SF_1, SF_2, R_3) for a SETAF SF and an extension $E_1 \in SF_1$ we define the* set of undecided links *w.r.t. E_1 as:*

$$\begin{aligned} U^{E_1}_{R_3} = \{(T, h) \in R_3 \mid & T \cap (E_1)^+_{R_1 \cup R_3} = \emptyset, \\ & \exists t \in T : t \in A_1 \setminus (E_1)^{\oplus}_{R_1}\}. \end{aligned}$$

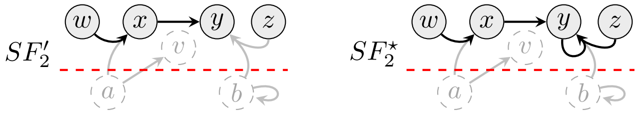
In what follows, we define the *modification*, which is applied on the reduct, and accounts for the effects of the undecided links. In particular, for each undecided link $(S, t) \in R_3$ we add to F_2' a (set-)self-attack from t (together with the F_2'-part of the attack) to itself.

Definition 10 (Modification)**.** *Let (SF_1, SF_2, R_3) be a splitting for a SETAF SF and E_1 an extension of SF_1. Take SF_2' as the (E_1, R_3)-reduct of SF_2 and $U^{E_1}_{R_3}$ as the set of undecided links w.r.t. E_1. We denote with $mod^{E_1}_{R_3}(SF_2') = SF_2^\star = (A_2^\star, R_2^\star)$ the $U^{E_1}_{R_3}$-modification (or simply modification) of SF_2' s.t. $A_2^\star = A_2'$ and $R_2^\star$ is given by:*

$$R_2' \cup \{((T \cap A_2') \cup \{h\}, h) \mid (T, h) \in U^{E_1}_{R_3},\ h \in A_2'\}.$$

Before we present the splitting theorem we illustrate Definitions 8–10 in the following example.

Example 2. *Consider the SETAF SF of Example 1 that separates arguments in $A_1 = \{a, b\}$ from those in $A_2 = \{v, w, x, y, z\}$. We select the preferred extension $E_1 = \{a\}$ of SF_1, and compute the reduct SF_2'. This is obtained by: (i) deleting v, as it is attacked by E_1; (ii) projecting $(\{a, w\}, x)$ to $(\{w\}, x)$ since $a \in E_1$; and (iii) neglecting the attack $(\{b, z\}, y)$ because $b \notin E_1$. Subsequently, we construct the modification $SF_2^\star$. For this, case (iii) is crucial: the attack $(\{b, z\}, y)$ is an* undecided link *since b is not in the range of E_1 and $\{b, z\}$ is not attacked by it. Thus, we introduce set-self-attack $(\{y, z\}, y)$ to SF_2'.*



Finally, we obtain $E_2 = \{w, z\}$ as the preferred extension of $SF_2^\star$, retrieving $E = E_1 \cup E_2 = \{a, w, z\}$ as a preferred extension of SF.

Having these notions at hand, we now establish the adequacy of the splitting technique for SETAFs. We start by establishing that (a) conflict-freeness of the sub-frameworks SF_1 and SF_2 carries over to the whole SETAF SF, and (b) conflict-free sets of SF induce conflict-free subsets in SF_1 and SF_2'.

Proposition 1. *Let (SF_1, SF_2, R_3) be a splitting for a SETAF $SF = (A, R)$ with $SF_1 = (A_1, R_1)$ and $SF_2 = (A_2, R_2)$. Let $SF_2' = SF_2^{E_1}$.*

1. *If $E_1 \in cf(SF_1)$ and $E_2 \in cf(mod^{E_1}_{R_3}(SF_2'))$, then $E_1 \cup E_2 \in cf(SF)$.*
2. *If $E \in cf(SF)$, then $E_1 = E \cap A_1 \in cf(SF_1)$ and $E \cap A_2 \in cf(SF_2')$.*

Finally, we are ready to characterize the splitting algorithm by generalising the splitting theorem for SETAFs under the standard Dung semantics.

Theorem 2. *Let (SF_1, SF_2, R_3) be a splitting for a SETAF $SF = (A, R)$ with $SF_1 = (A_1, R_1)$, $SF_2 = (A_2, R_2)$, and $\sigma \in \{stb, adm, com, pref, grd\}$. Further, let $SF_2^\star = mod^{E_1}_{R_3}(SF_2')$.*

1. *If $E_1 \in \sigma(SF_1)$ and $E_2 \in \sigma(SF_2^\star)$, then $E_1 \cup E_2 \in \sigma(SF)$.*
2. *If $E \in \sigma(SF)$, then $E_1 = E \cap A_1 \in \sigma(SF_1)$ and $E \cap A_2 \in \sigma(SF_2^\star)$.*

While the existing instantiation procedure from ABA frameworks to SETAFs provides a foundation for defining splitting, attempting to directly replicate the SETAF-style idea of splitting among assumptions fails to yield a natural notion of splitting. This disconnect stems from a fundamental structural difference: in SETAFs, attacks are primitive, whereas in ABA, they are derived from the underlying deductive system $(\mathcal{L}, \mathcal{R})$. As a result, naively mimicking SETAF-style splitting in ABA would require (i) arbitrarily partitioning the assumption set into $\mathcal{A}_1$ and $\mathcal{A}_2$, and (ii) computing attacks as derivations from assumptions in $\mathcal{A}_1$ to those in $\mathcal{A}_2$. However, splitting should be possible solely by inspecting the knowledge base at hand. Moreover, while instantiating ABAFs into SETAFs has been shown useful in specific contexts (Russo, Rapberger, and Toni 2024; Buraglio et al. 2024b), this approach comes with a critical drawback: it can yield an exponential growth in the number of collective attacks generated, thus increasing in input size. This inefficiency motivates many ABA solvers to operate directly on ABAFs rather than relying on their abstract representations. Therefore, to enable an efficient form of splitting, we propose a dedicated splitting algorithm tailored to the syntactic structure of ABAFs.

4 Splitting in Assumption-Based Argumentation

In this section we present splitting results for ABAFs. The rule-set of an ABAF is split into a bottom and a top part whenever no assumption occurs in the bottom part whose contrary is derived by some rule in the top. This ensures that the assumptions in the bottom can be evaluated independently of what can be deduced by inspecting the top part. We capture this intuition via the notion of splitting set:

Definition 11. *Given an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, a set $S \subseteq \mathcal{L}$ is a splitting set (or simply a splitting) of D if $S = atom(S)$ and for all $r \in \mathcal{R}$, $head(r) \in S$ implies $body(r) \subseteq S$.*

A splitting set partitions the deductive system into two sub-systems $(\mathcal{L}_1, \mathcal{R}_1)$ and $(\mathcal{L}_2, \mathcal{R}_2)$, called the 'bottom' and 'top'. In particular, we have (i) $\mathcal{L}_1 = S$ and $\mathcal{R}_1 = \{r \in \mathcal{R} \mid head(r) \in S\}$ and (ii) $\mathcal{L}_2 = \mathcal{L} \setminus S$ and $\mathcal{R}_2 = \{r \in \mathcal{R} \mid head(r) \notin S\}$. These induce respectively $D_1 = (\mathcal{L}_1, \mathcal{R}_1, \mathcal{A}_1, ^{-1})$ and $D_2 = (\mathcal{L}_2, \mathcal{R}_2, \mathcal{A}_2, ^{-2})$ with $\mathcal{A}_i = \mathcal{L}_i \cap \mathcal{A}$ and $^{-i}$ defined over $\mathcal{A}_i$.

Example 3. *Consider again the ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ from Example 1, with assumptions $\mathcal{A} = \{a, b, v, w, x, y, z\}$, sentences $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}} \cup \{p\}$, and rules $\mathcal{R}$ as follows:*

$$\overline{b} \leftarrow b \quad p \leftarrow a \quad \overline{v} \leftarrow a \quad \overline{x} \leftarrow w, p \quad \overline{y} \leftarrow x \quad \overline{y} \leftarrow \overline{b}, z$$

Take the set $S = \{a, b, \overline{a}, \overline{b}, p\}$. It can be easily checked that S is a splitting set of D, through which we obtain two subsystems $(\mathcal{L}_1, \mathcal{R}_1)$ (bottom) and $(\mathcal{L}_2, \mathcal{R}_2)$ (top) such that:

- *$\mathcal{L}_1 = S$ and $\mathcal{L}_2 = \mathcal{L} \setminus S$;*
- *$\mathcal{R}_1 = \{\overline{b} \leftarrow b,\ p \leftarrow a\}$ and $\mathcal{R}_2 = \mathcal{R} \setminus \mathcal{R}_1$.*

Notice that some atoms contained in $\mathcal{L}_1$ (but not in $\mathcal{L}_2$) may occur in the body of some rule in $\mathcal{R}_2$ (a, $\overline{b}$ and p in Example 3). This intermediate mismatch will be resolved later by the notion of *reduct*. Moreover, their occurrences in $\mathcal{R}_2$ does not affect the acceptance status of such atoms. In fact, a first sanity check, we observe that our notion of splitting prevents building attacks from assumptions of D_2 towards assumptions of D_1 using top-rules in $\mathcal{R}_2$. This is ensured by the fact that contraries of assumptions occurring in the bottom part are not derived via rules in the top part (via construction of $\mathcal{R}_2$). As a result, assumptions in $\mathcal{A}_1$ are attacked only via rules in $\mathcal{R}_1$ by assumptions in $\mathcal{A}_1$. Thus, no attack generated from $\mathcal{A}_2$ (by means of rules in $\mathcal{R}_2$) is directed towards $\mathcal{A}_1$.

Proposition 2. *Let D be an ABAF and S a set of literals that splits D into D_1 and D_2. For every derivation $T \vdash^R \overline{a}$ with $a \in \mathcal{A}_1$, it holds that $R \subseteq \mathcal{R}_1$ and $T \subseteq \mathcal{A}_1$.*

The attacks of the bottom part can be extended in a conservative way: whatever happens in the second sub-framework does not affect the acceptability of assumptions in D_1. Thus, to compute incrementally an extension of an ABAF D, we can first select an extension E of D_1 and later modify D_2 according to the information contained in E. Consequently, we can evaluate the modified framework D_2 and augment its extensions with E. Again, we follow the approach of Baumann and appeal to the notions of *reduct* and *modification* in application to D_2 via a two-step process. First, we propagate all the information we get from a σ-extension E of D_1 to ensure that rules in contrast with E are removed. The outcome is called the E-reduct of D_2.

Definition 12. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF, S a splitting set of D into D_1 and D_2 and E a σ-extension of D_1. We call $D_2^E = (\mathcal{L}_2, \mathcal{R}_2^E, \mathcal{A}_2, ^{-2})$ the E-reduct (or simply reduct) of D_2, where $\mathcal{R}_2^E$ is obtained by deleting:*

- *each rule $r \in \mathcal{R}_2$ with $body(r) \cap S \not\subseteq Th_{D_1}(E)$;*
- *all literals in $Th_{D_1}(E)$ from the remaining rules.*

As we anticipated, the rule-set $\mathcal{R}_2^E$ now contains all and only those atoms occurring in $\mathcal{L}_2$. Therefore, the reduct can be evaluated in complete isolation from D_1. In the second step, we modify the reduct to propagate the information about assumptions (or their contraries) which are not contained in $Th_{D_1}(E)$. We call these assumptions *undecided*, as they are not in E nor their contrary is derivable from it (i.e. are not attacked by E). Then, the *set of undecided assumptions* of D_1 w.r.t. E is $\mathsf{UA}_{D_1}(E) = \{a \in \mathcal{A}_1 \mid a \notin E \text{ and } \overline{a} \notin Th_{D_1}(E)\}$. Since their status can be transmitted to other assumptions via rules, we define the concept of *undecided theory* of D_1, capturing all sentences derived from some undecided (but no defeated) assumptions.

Definition 13. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF and $E \in \sigma(D)$. The* undecided theory $\mathsf{UT}_D(E)$ *of D w.r.t. E is the set of sentences $p \in \mathcal{L}$ for which there is a $T \subseteq \mathcal{A}$ such that (i) $T \vdash p$, (ii) $T \cap \mathsf{UA}_D(E) \neq \emptyset$, and (iii) $\overline{T} \cap Th_D(E) = \emptyset$.*

Rules in D_2 whose bodies contain elements of $\mathsf{UT}_{D_1}(E)$ might carry over undecidedness from D_1. However, this scenario could be overwritten by the presence of *incompatible sentences* w.r.t. E, captured by $\mathsf{IS}_{D_1}(E) = Th_{D_1}(E^+_{\mathcal{R}_1}) \cup \overline{E}$, where $E^+_{\mathcal{R}_1} = \{a \in \mathcal{A}_1 \mid E \vdash^R \overline{a}, R \subseteq \mathcal{R}_1\}$. Hence, a set of sentences from D_1 will carry undecidedness to sentences in D_2 if and only if (i) none of its elements is incompatible and (ii) at least one of its elements is in the undecided theory w.r.t. the previously selected extension. This concept mirrors the notion of undecided links for SETAFs.

We are now in the position to formally define the *modification* of D_2^E. First, we expand the set of sentences with a fresh assumption x_u and corresponding contrary. Further, we introduce (i) a loop-rule for x_u and (ii) a modified version of every rule with some undecided (but no incompatible) sentence in the body. In particular, we expand their body with x_u, after projecting to $\mathcal{L}_2$.

Definition 14. *Let D be an ABAF, S a set that splits D into two sub-frameworks D_1 and D_2 and E an extension of D_1. Further, let D_2' be the E-reduct of D_2. We use $mod_{D_1}^E(D_2') = D_2^\star = (\mathcal{L}_2^\star, \mathcal{R}_2^\star, \mathcal{A}_2^\star, ^{-\star})$ to denote the E-modification (or simply modification) of D_2' where $D_2^\star = D_2'$ if $\mathsf{UA}_{D_1}(E) = \emptyset$, and otherwise:*

$\mathcal{L}_2^\star = \mathcal{L}_2 \cup \{x_u, \overline{x_u}\}$;

$\mathcal{R}_2^\star = \mathcal{R}_2' \cup \{\overline{x_u} \leftarrow x_u\} \cup$
$\{head(r) \leftarrow (body(r) \cap \mathcal{L}_2) \cup \{x_u\} \mid r \in \mathcal{R}_2,$
$body(r) \cap \mathsf{IS}_{D_1}(E) = \emptyset, body(r) \cap \mathsf{UT}_{D_1}(E) \neq \emptyset\}$.

Example 4. *Consider again the ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ from Example 3 and splitting set $S = \{a, b, \overline{a}, \overline{b}, p\}$. We know that $\{a\} \in pref(D_1)$. Therefore, the $\{a\}$-reduct of D_2 is $D_2^{\{a\}} = (\mathcal{L}_2, \mathcal{R}_2^{\{a\}}, \mathcal{A}_2, ^{-2})$, where the rule-set $\mathcal{R}_2^{\{a\}}$ is:*

$$\overline{v} \leftarrow a \qquad \overline{x} \leftarrow w, p \qquad \overline{y} \leftarrow x \qquad \overline{y} \leftarrow \overline{b}, z$$

Moreover, $\mathsf{UA}_{D_1}(\{a\}) = \{b\}$ *is the set of undecided assumptions and* $UT_{D_1}(\{a\}) = \{b, \bar{b}\}$. *We then compute the modification* $D_2^\star$ *w.r.t.* $\{a\}$ *by expanding the set of sentences* $\mathcal{L}_2$ *with* $\{x_u, \overline{x_u}\}$ *and the rule-set* $\mathcal{R}_2^{\{a\}}$ *above such that:*

$$\overline{v} \leftarrow \quad \overline{x} \leftarrow w \quad \overline{y} \leftarrow x \quad \overline{y} \leftarrow x_u, z \quad \overline{x_u} \leftarrow x_u$$

Since $\{w, z\} \in$ *pref*$(D_2^\star)$*, we get* $\{a, w, z\}$ *as in Example 2.*

We can now prove that our procedure preserves conflict-free sets under incremental computation as well as projection to sub-frameworks, as for SETAFs in Section 3.

Proposition 3. *Let* $S \subseteq \mathcal{L}$ *be a splitting set of an ABAF* D *into* D_1 *and* D_2*. Moreover, let* $D_2^\star = mod_{D_1}^{E_1}(D_2^{E_1})$.

1. *If* $E_1 \in$ *cf*(D_1) *and* $E_2 \in$ *cf*$(D_2^\star)$*, then* $E_1 \cup E_2 \in$ *cf*(D).
2. *If* $E \in$ *cf*(D)*, then* $E \cap \mathcal{A}_1 \in$ *cf*(D_1) *and* $E \cap \mathcal{A}_2 \in$ *cf*(D_2^E).

We then show that splitting is adequate with respect to stable, admissible, complete, preferred and grounded semantics. Due to space constraints we include proof details only for admissible semantics, being prototypical for the others.

Theorem 3. *Let* S *be a splitting set for an ABAF* D *into* D_1 *and* D_2 *and* $\sigma = \{$*stb, adm, com, pref, grd*$\}$*. Further, let* $mod_{D_1}^{E_1}(D_2^{E_1}) = D_2^\star$.

1. *If* $E_1 \in \sigma(D_1)$ *and* $E_2 \in \sigma(D_2^\star)$*, then* $E_1 \cup E_2 \in \sigma(D)$.
2. *If* $E \in \sigma(D)$*, then* $E \cap \mathcal{A}_1 \in \sigma(D_1)$ *and* $E \cap \mathcal{A}_2 \in \sigma(D_2^\star)$.

Proof. **(admissible)**. (1.) Since admissibility implies conflict-freeness from Proposition 3, we know that $E = E_1 \cup E_2 \in$ *cf*(D). Thus we only need to show that E defends itself in D, i.e. for all $a \in E$, if $T \vdash \overline{a}$, then $T' \vdash \overline{t}$ for some $t \in T$ and $T' \subseteq E$. If $a \in E_1$, we know that a is defended by E_1 in $\mathcal{A}_1$ from hypothesis. Thus, from Proposition 2, we can deduce that $E_1 \in$ *adm*(D). Consider now an assumption $a \in E_2$ and some $T \subseteq \mathcal{A}$ such that $T \vdash^R \overline{a}$ and $R \subseteq \mathcal{R}$. If $\overline{T} \cap Th_{D_1}(E_1) \neq \emptyset$, then E_1 defends a against T in D. If $\overline{T} \cap Th_{D_1}(E_1) = \emptyset$, this means that $T \subseteq \mathcal{A}_2$ and $T \vdash \overline{a}$ in D_2' (a is attacked in the reduct) or $T \cup x_u \vdash^R \overline{a}$ in $D_2^\star$ (a is attacked in the modification). In both cases, since E_2 is conflict-free and defends a in $D_2^\star$, there is a $T' \subseteq E_2$ such that $T' \vdash \overline{t}$ with $t \in T$. We distinguish two cases: either (i) $T' \vdash \overline{t}$ already in D_2, in which case a is defended by E in D, or (ii) there is some $T'' \supset T'$ such that $T'' \vdash \overline{t}$ in D and $T'' \cap \mathcal{A}_1 \subseteq E_1$. Thus, since $T \subseteq E_1 \cup E_2$, a is defended by $E_1 \cup E_2$ in D. In any case a is defended in D by E.

(2.) By Proposition 3, we get $E_1 = E \cap \mathcal{A}_1 \in$ *cf*(D_1) and $E_2 = E \cap \mathcal{A}_2 \in$ *cf*(D_2'). First, we know that $E_1 = E \cap \mathcal{A}_1 \in$ *adm*(D_1) because E defends itself in D and $E \cap \mathcal{A}_1$ is not attacked by a subset of $\mathcal{A}_2$ (Proposition 2). It remains to prove that $E_2 = E \cap \mathcal{A}_2 \in$ *adm*$(D_2^\star)$. Take an assumption $a \in E_2$ such that $T \vdash \overline{a}$ in $D_2^\star$. Each such derivation corresponds to exactly one derivation $T' \vdash^R \overline{a}$ with $R \subseteq \mathcal{R}$. There are two cases: either (i) $T' = T \subseteq \mathcal{A}_2$ and $R \subseteq \mathcal{R}_2$ or (ii) $T' \supset T \setminus \{x_u\}$ where $T' \setminus T \subseteq E_1$ (assumptions deleted from simplified rules in the reduct). From both (i) and (ii) we deduce that $\overline{T'} \cap Th_{D_1}(E_1) = \emptyset$: for (i) because it would entail $T \not\subseteq \mathcal{A}_2$; for (ii) because otherwise $T \not\vdash \overline{a}$ in $D_2^\star$. Nonetheless, since E defends a in D, in case (i) there is a counter-attack $T'' \vdash^{\mathcal{R}_2} \overline{t}$ such that $T'' \subseteq E$ and $t \in (T \setminus \{a\})$. In case (ii), the same holds but $t \in (T' \setminus \{a, x_u\})$. If $T'' \cap \mathcal{A}_1 = \emptyset$, we know that $\{t\} \subseteq \mathcal{A}_2'$ and together with the fact that $T'' \subseteq E$, we derive $T'' \subseteq E \cap \mathcal{A}_2' = E_2$. Hence, T'' defends a from T in $D_2^\star$. If $T'' \cap \mathcal{A}_1 \neq \emptyset$, then $T'' \cap \mathcal{A}_1 \subseteq E_1$. Therefore, from $T'' \subseteq E$ we get $T'' \cap \mathcal{A}_2' \vdash^{\mathcal{R}_2'} \overline{t}$, which defends a against T in $D_2^\star$. Thus a is always defended in $D_2^\star$, as desired. ☐

The above results guarantees the correctness of our divide-and-conquer algorithm. However, as anticipated in the introduction, splitting is only possible when certain structural properties are met. For graph-based frameworks, it is required that all attacks between the two sub-frameworks share the same direction. For this reason, the presence of a single attack can invalidate the possibility of splitting (e.g. when the AF in question has the form of a Strongly Connected Component (SCC)). In the case of ABA, the same can happen when an assumption b in the splitting set S is attacked by assumptions outside of it. In such cases, splitting is impossible, as the bottom part of ABAF is not independent from the top. Thus, the acceptability of b cannot be ensured by projecting to the bottom. Similarly to graph-based frameworks, the possibility to make use of a splitting algorithm may be hindered, even in cases where only one out of many assumptions in S is externally attacked. To overcome such issue, we consider in the following section a generalised version of ABA splitting.

5 Parametrised Splitting

We now introduce a more general version of splitting for ABAFs, called *parametrised splitting* inspired by Baumann et al. (2012). This relaxes the structural constraint for the application of splitting, potentially allowing a fixed number of assumptions in the bottom part to be attacked from assumptions in the top. Precisely, in contrast with the previous notion of splitting, we allow some contraries of assumptions occurring in bodies of $\mathcal{R}_1$ to appear as the heads of rules in $\mathcal{R}_2$. The number of these assumptions then represents a measure of how far we are from obtaining a splitting of ABA knowledge base. The concept of a splitting set is then generalised accordingly as follows:

Definition 15. *For any ABAF* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$*, a set* $S \subseteq \mathcal{L}$ *is called a* quasi-splitting *of* D *if* $S = atom(S)$ *and for all* $r \in \mathcal{R}$*,* $head(r) \in S$ *implies* $body(r) \setminus \mathcal{A} \subseteq S$*. Let* $V_S^{\leftarrow} = \{b \in \mathcal{A} \setminus S \mid \exists r, r' \in \mathcal{R} : b \in body(r) \cap \mathcal{A}, head(r) \in S, head(r') = \bar{b}, r \neq r'\}$*. We call* S*:*

- *k-splitting of* D*, if* $|V_S^{\leftarrow}| = k$*;*
- *(proper) splitting of* D*, if* $|V_S^{\leftarrow}| = 0$.

As before, the rule-set is split into a bottom and top part, depending on the rule-head respectively being or not in S. As a result, $V_S^{\leftarrow}$ is the set of assumptions in the bottom whose contrary is derived in the top. We call $V_S^{\leftarrow}$ the set of *vulnerabilities* with respect to S, since it contains assumptions that are attacked by S. Whenever $|V_S^{\leftarrow}| \neq 0$, there are some heads in $\mathcal{R}_2$ whose corresponding assumption may appear in bodies of $\mathcal{R}_1$. Therefore, the notion of splitting of Definition 11 corresponds to a 0-splitting. To account for elements of $V_S^{\leftarrow}$, the ABAFs D_1 and D_2 induced by the chosen splitting set are constructed in a slightly different way than before. In particular, we fix D_1 and D_2 as before, but let $\mathcal{L}_1 = S \cup V_S^{\leftarrow} \cup \overline{V_S^{\leftarrow}}$. Moreover, since contraries in $\overline{V_S^{\leftarrow}}$ may be derived by top rules, the status of their corresponding assumptions in the bottom depends on rules in the top. Consequently D_1 cannot be evaluated in complete isolation from the rest, in contrast with proper splitting.

For computing extension of the sub-framework D_1, we first need to modify the ABAF. First, we modify the rules by removing body-atoms not in $\mathcal{L}_1$. Indeed, these atoms occur in $\mathcal{L}_2$ and are unattacked in D, therefore they can be disregarded when evaluating D_1. Further, we proceed by adding: (i) a fresh assumption b' (and its contrary $\overline{b'}$) for each $b \in V_S^{\leftarrow}$; (ii) rules that encode the choice for or against the presence of each assumption $b \in V_S^{\leftarrow}$ in the extension. In this way, we store at the object level the meta-information regarding our choices on each $b \in V_S^{\leftarrow}$.

Definition 16. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF, $S \subseteq \mathcal{L}$ be a quasi-splitting of D inducing the sub-frameworks D_1 and D_2. Moreover, let $V_S^{\leftarrow}$ be the set of vulnerabilities of D_1 with respect to S and $(\mathcal{R}_1)_{\downarrow \mathcal{L}_1} = \{head(r) \leftarrow body(r) \cap \mathcal{L}_1 \mid r \in \mathcal{R}_1\}$. From D_1 we construct the ABAF $\llcorner D_1 \lrcorner = (\llcorner \mathcal{L}_1 \lrcorner, \llcorner \mathcal{R}_1 \lrcorner, \llcorner \mathcal{A}_1 \lrcorner, ^-)$ by letting:*

- $\llcorner \mathcal{L}_1 \lrcorner = \mathcal{L}_1 \cup \{b', \overline{b'} \mid b \in V_S^{\leftarrow}\}$;
- $\llcorner \mathcal{R}_1 \lrcorner = (\mathcal{R}_1)_{\downarrow \mathcal{L}_1} \cup \{\overline{b} \leftarrow b', \overline{b'} \leftarrow b \mid b \in V_S^{\leftarrow}\}$.

Intuitively, the additional rules allow us to choose whether we want to accept an extension E of $\llcorner D_1 \lrcorner$ containing b or one that does not. After this choice, we can safely compute the E-reduct of D_2, as for proper splitting. In this way, we propagate the meta-information to which we committed by means of our choice. A further modification of D_2 is now needed to ensure our hypothesis regarding b: we add a fact-rule $b \leftarrow$ or a loop-rule $\overline{b} \leftarrow b$, depending on whether the previously chosen extension E contains b or b'. These represent a type of (positive and negative) constraints in ABA.

Definition 17. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF, S a quasi-splitting of D into D_1 and D_2. Moreover, let $V_S^{\leftarrow}$ be the set of vulnerabilities with respect to S and D_2^E the E-reduct of D_2 for some $E \in \sigma(\llcorner D_1 \lrcorner)$. We denote with $\ulcorner D_2^E \urcorner = (\mathcal{L}_2, \ulcorner \mathcal{R}_2^E \urcorner, \mathcal{A}_2, ^{-2})$ the ABAF such that:*

$$\ulcorner \mathcal{R}_2^E \urcorner = \mathcal{R}_2^E \cup \{b \leftarrow \mid b \in E \cap V_S^{\leftarrow}\} \cup \{\overline{b} \leftarrow b \mid b' \in E\}.$$

Notice that such a modification can make $\ulcorner D_2^E \urcorner$ non-flat, as $cl(\emptyset) = \{b \mid b \in E \cap V_S^{\leftarrow}\}$. For stable semantics, however, this does not result in a higher complexity for the same reasoning tasks (Cyras, Heinrich, and Toni 2021).

Example 5. *Consider the ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ where $\mathcal{A} = \{a, b, c, d\}$, $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}} \cup \{p\}$, and rule-set $\mathcal{R}$:*

$$\overline{b} \leftarrow a \qquad \overline{d} \leftarrow b \qquad \overline{a} \leftarrow p, c \qquad p \leftarrow b$$

First, $E = \{b, c\}$ and $E' = \{a, c, d\}$ are stable extensions in D. Now let $S = \{a, \overline{a}, d, \overline{d}, p\}$ be a quasi-splitting of D and $V_S^{\leftarrow} = \{b\}$ the set of vulnerabilities w.r.t. S. We get $\llcorner \mathcal{L}_1 \lrcorner = S \cup \{b\} \cup \{\overline{b}\} \cup \{b', \overline{b'}\}$ and $\llcorner \mathcal{R}_1 \lrcorner$ such that:

$$\overline{d} \leftarrow b \qquad \overline{a} \leftarrow p, c \qquad p \leftarrow b \qquad \overline{b'} \leftarrow b \qquad \overline{b} \leftarrow b'$$

We derive two stable extensions $E_1 = \{b\}$ and $E_1' = \{b', a, d\}$. Now consider D_2 with $\mathcal{L}_2 = \mathcal{L} \setminus S = \{b, c, \overline{b}, \overline{c}\}$. For the former we get $\ulcorner D_2^{E_1} \urcorner$ with $\ulcorner \mathcal{R}_2^{E_1} \urcorner = \emptyset \cup \{b \leftarrow\}$ from which we derive $E_2 = \{b, c\}$ as a stable extension. For the latter we get $\ulcorner D_2^{E_1'} \urcorner$ with $\ulcorner \mathcal{R}_2^{E_1} \urcorner = \{\overline{b} \leftarrow\} \cup \{\overline{b} \leftarrow b\}$ from which we derive $E_2' = \{c\}$ as a stable extension. We then obtain $E = (E_1 \cap S) \cup E_2$ and $E' = (E_1' \cap S) \cup E_2'$.

Theorem 4. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF and $S \subseteq \mathcal{L}$ a quasi-splitting inducing the sub-frameworks D_1 and D_2.*

1. *If $E_1 \in stb(\llcorner D_1 \lrcorner)$ and $E_2 \in stb(\ulcorner D_2^{E_1} \urcorner)$, then $(E_1 \cap S) \cup E_2 \in stb(D)$.*
2. *If $E \in stb(D)$, then there is a set $X \subseteq \{a' \mid a \in V_S^{\leftarrow}\}$ s.t. $E_1 = (E \cap S) \cup X \in stb(\llcorner D_1 \lrcorner)$ and $E_2 = E \cap \mathcal{A}_2 \in stb(\ulcorner D_2^{E_1} \urcorner)$.*

Proof. In what follows, let $D_2' = (\mathcal{L}_2, \mathcal{R}_2', \mathcal{A}_2, ^{-2})$ be the reduct of D_2 w.r.t. $E_1 = (E \cap S) \cup X$ and $E = E_1 \cup E_2$.

(1.) To prove the statement we need to show $(E_1 \cap S) \cup E_2 \in cf(D)$ and $((E_1 \cap S) \cup E_2)^{\oplus}_{\mathcal{R}} = \mathcal{A}$. We start with conflict-freeness. Since $E_1 \in cf(\llcorner D_1 \lrcorner)$, then $E_1 \cap S \in cf(\llcorner D_1 \lrcorner)$ (less assumptions) and $E_1 \cap S \in cf(D_1)$ (less attacks). We show that $E_1 \cap S \in cf(D)$. Assume towards contradiction that $E_1 \cap S \vdash^R \overline{a}$ for some $a \in E_1 \cap S$ and $R \subseteq \mathcal{R}$. Since $E_1 \cap S \in cf(D_1)$ and $\mathcal{R}_2 = \{r \in \mathcal{R} \mid head(r) \notin S\}$, it must be that $E_1 \cap S \vdash^{R_2} p$ where $R_2 = R \cap \mathcal{R}_2$ for some $p \in body(r')$ and rule $r' \in R_1 = R \cap \mathcal{R}_1$. Since $p \in \mathcal{L}_2$, we distinguish two cases: if $p \in \overline{V_S^{\leftarrow}}$, then it is derivable from some b' in $\llcorner D_1 \lrcorner$, and r' fires; if $p \notin \overline{V_S^{\leftarrow}}$, then it gets removed from the body of r' in $\llcorner D_1 \lrcorner$, which in turn fires. Thus, in both cases we have that $E_1 \cap S \vdash^{R_1'} \overline{a}$ where $R_1' \subseteq \llcorner \mathcal{R}_1 \lrcorner$. Contradiction. Consider now $E_2 \in stb(\ulcorner D_2' \urcorner)$. Since being stable implies conflict-freeness we immediately get $E_2 \in cf(\ulcorner D_2' \urcorner)$. Again, since $\mathcal{R}_2' \subseteq \ulcorner \mathcal{R}_2' \urcorner$, we obtain $E_2 \in cf(D_2')$. Furthermore, Proposition 3 for proper splittings, together with $E_1 \cap S \in cf(D_1)$ and $E_2 \in cf(D_2')$, entail $(E_1 \cap S) \cup E_2 \nvdash^R \overline{a}$ for any $a \in E_2$ and $R \subseteq \mathcal{R}$. It only remains to consider possible attacks from E_2 to $E_1 \cap S$ in D. Suppose that there are $T \subseteq E_2$ and $a \in E_1 \cap S$ such that $T \vdash^R \overline{a}$ for some $R \subseteq \mathcal{R}$. First, notice that since $T \subseteq E_2$, we get $body(R) \cap S = \emptyset \subseteq Th_{D_1}(E_1)$, and thus $R \subseteq \mathcal{R}_2'$. Moreover, $a \in V_S^{\leftarrow}$ so that $\ulcorner \mathcal{R}_2' \urcorner = \mathcal{R}_2' \cup \{a \leftarrow\}$. Therefore, $T \vdash^R a$ and $T \vdash^R \overline{a}$ for some $R \subseteq \ulcorner \mathcal{R}_2' \urcorner$, i.e. E_2 is either not conflict-free or not closed in $\ulcorner D_2' \urcorner$. We now show that $((E_1 \cap S) \cup E_2)^{\oplus}_{\mathcal{R}} = \mathcal{A}$. Towards contradiction, consider an assumption $a \notin ((E_1 \cap S) \cup E_2)^{\oplus}_{\mathcal{R}}$. Assume $a \in S$. By hypothesis, $E_1 \in stb(\llcorner D_1 \lrcorner)$, i.e. either $a \in E_1$ or $E_1 \vdash^R \overline{a}$ for some $R \subseteq \llcorner \mathcal{R}_1 \lrcorner$. From our assumption, we get $a \notin (E_1 \cap S)^{\oplus}_{\mathcal{R}}$, that is (i) $a \notin E_1 \cap S$ and (ii) $E_1 \cap S \nvdash^R \overline{a}$ for any $R \subseteq \mathcal{R}$. If (i) holds, we immediately derive $E_1 \vdash^R \overline{a}$ for some $R \subseteq \llcorner \mathcal{R}_1 \lrcorner$. Consider now our assumption (ii). Because $a \in S$ we know that every rule of R is contained in $\mathcal{R}_1$. For the same reason such rules are in $\llcorner\mathcal{R}_1\lrcorner$ ($b \notin V_S^{\leftarrow}$). Therefore, $E_1 \cap S \nvdash^R \overline{a}$ for any $R \subseteq \llcorner\mathcal{R}_1\lrcorner$ in contradiction with our hypothesis. Assume now $a \in \mathcal{A} \setminus S$. By hypothesis we know either $a \in E_2$ or $E_2 \vdash^R \overline{a}$ for some $R \subseteq \ulcorner\mathcal{R}_2'\urcorner$. From the assumption, we get $a \notin (E_2)^{\oplus}_{\mathcal{R}}$, that is (i) $a \notin E_2$ and (ii) $E_2 \nvdash^R \overline{a}$ for any $R \subseteq \mathcal{R}$. As before, from (i) and our hypothesis we derive $E_2 \vdash^R \overline{a}$ must hold for some $R \subseteq \ulcorner\mathcal{R}_2'\urcorner$. If $a \in V_S^{\leftarrow}$, there are two possibilities: $a \in E_1 \setminus S$ or $a \notin E_1 \setminus S$. In the first scenario, $\ulcorner\mathcal{R}_2'\urcorner = \mathcal{R}_2' \cup \{a \leftarrow\}$. Again, $E_2 \vdash^R a$ and $E_2 \vdash^R \overline{a}$ for some $R \subseteq \ulcorner\mathcal{R}_2'\urcorner$, in contradiction with the fact that E_2 is a stable extension of $\ulcorner D_2'\urcorner$. If $a \notin E_1 \setminus S$, then $a' \in E_1$, which means $\ulcorner\mathcal{R}_2'\urcorner = \mathcal{R}_2' \cup \{\overline{a} \leftarrow a\}$. Since $a \notin E_2$, the loop-rule $\overline{a} \leftarrow a$ is not in R, therefore $R \subseteq \mathcal{R}_2'$. Thus, for each rule $r' \in \mathcal{R}_2'$ there is a rule $r \in \mathcal{R}_2$ such that $body(r) \subseteq body(r) \cup Th_{D_1}(E_1)$. Hence, it follows directly that $(E_1 \cap S) \cup E_2 \vdash^R \overline{a}$ for some $R \subseteq \mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}$. If $a \notin V_S^{\leftarrow}$, then $a \notin \mathcal{A}_1$. If $E_2 \vdash^R \overline{a}$ for some $R \subseteq \ulcorner\mathcal{R}_2'\urcorner$, it is not because $\{\overline{a} \leftarrow a\} \subseteq \ulcorner\mathcal{R}_2'\urcorner$. Thus $R \subseteq \mathcal{R}_2'$. As before, for each rule $r' \in \mathcal{R}_2'$ there is exactly one rule $r \in \mathcal{R}_2$ such that $body(r) \subseteq body(r') \cup Th_{D_1}(E_1 \cap S)$. As a result, in D it holds that $(E_1 \cap S) \cup E_2 \vdash \overline{a}$. Contradiction.

(2.) First we get $E \in cf(D)$ and thus $E \cap S \in cf(D_1)$ (less attacks). Now let $B = \mathcal{A}_1 \setminus (E \cap S)^{\oplus}_{\mathcal{R}}$. Since $E \in stb(D)$, it attacks every other assumption. Hence, we can infer that assumptions in B are contained in $\mathcal{A}_1$ and attacked by $E \cap \mathcal{A}_2$ in D, that is $B \subseteq E^+_{\mathcal{R}} \setminus (E \cap S)^{\oplus}_{\mathcal{R}} = (E \cap \mathcal{A}_2)^+_{\mathcal{R}}$. Therefore there is a rule $r \in \mathcal{R}_2$ with $head(r) = \overline{b}$ for each $b \in B$, meaning that $B = V_S^{\leftarrow}$. Now let $X = \{b' \mid b \in B\}$. Thus $\ulcorner\mathcal{R}_2'\urcorner$ contains a pair of rule $\{\overline{b} \leftarrow b', \overline{b'} \leftarrow b\}$ for each $b \in B$. Consequently, conflict-freeness of $(E \cap S) \cup X$ is ensured since $b \notin E \cap S$ for all $b \in B$. Moreover, X attacks every $b \in B$ in $\llcorner D_1\lrcorner$, making $(E \cap S) \cup X$ stable. It now remains to show $E_2 = E \cap \mathcal{A}_2 \in stb(\ulcorner D_2'\urcorner)$. As before, we know that $E \cap \mathcal{A}_2 \in cf(D_2)$ since E is conflict-free in D (less assumptions), and $E \cap \mathcal{A}_2 \in cf(D_2')$ because $Th_{D_2'}(E \cap \mathcal{A}_2) \subseteq Th_{D_2}(E \cap \mathcal{A}_2)$ (less rules and attacks). Consider now the modified framework $\ulcorner D_2'\urcorner$ wrt $(E \cap S) \cup X$. By construction, $E \cap \mathcal{A}_2 \notin cf(\ulcorner D_2'\urcorner)$ only if $b \in B \cap (E \cap \mathcal{A}_2)$. Recall that $B \subseteq (E \cap \mathcal{A}_2)^+_{\mathcal{R}}$. Thus, $E \cap \mathcal{A}_2 \notin cf(D)$. By contradiction, we derive that $E \cap \mathcal{A}_2$ is conflict free in $\ulcorner D_2'\urcorner$. We now show that $E_2 \vdash^R \overline{a}$ for all $a \in \mathcal{A}_2 \setminus E_2$ and some $R \subseteq \ulcorner\mathcal{R}_2'\urcorner$. Towards contradiction, we assume there is an $a \in \mathcal{A}_2 \setminus E_2$ such that $E_2 \nvdash^R \overline{a}$, i.e. $\overline{a} \notin Th_{\ulcorner D_2'\urcorner}(E_2)$. Therefore, since $a \notin E_2$, we get $\overline{a} \notin Th_{D_2'}(E_2)$. Hence, before the reduct is applied, it holds that $(E \cap S) \cup E_2 \nvdash^R \overline{a}$ with $R \subseteq \mathcal{R}_2$. Since no rule $r \in \mathcal{R}_1$ is such that $head(r) = \overline{a}$, we derive $(E \cap S) \cup E_2 \nvdash^R \overline{a}$ in D, contradicting our hypothesis. Finally, we ensure that $cl(E_2) = E_2$ in $\ulcorner D_2'\urcorner$. Assume the contrary holds. Since D_2' is flat, that means $\{a \leftarrow\} \subseteq \ulcorner\mathcal{R}_2'\urcorner$ and $a \notin E_2$. These facts respectively entail $a \in E_1$ and $E_2 \vdash \overline{a}$, against the conflict-freeness of E. Thus, E_2 is conflict-free, closed and attacks every other assumption. $\square$

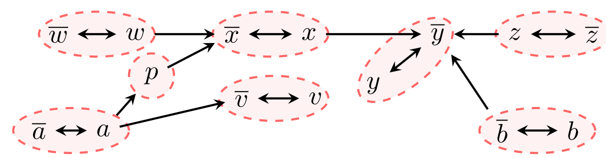
6 Computing the Splitting Set

In order to apply the introduced splitting schemes, we need to first compute a splitting of the ABAF and SETAF respectively. In this section, we show that this can be reduced to the same graph problem as for computing a splitting of an AFs, and we can therefore reuse existing algorithms (Baumann, Brewka, and Wong 2011) for the computation of our splittings. First, for a SETAF SF we consider its primal graph, which draws an egde from argument a to b if there is an attack $(T, b) \in SF$ with $a \in T$. For ABA splittings, we build a Dependency Graph G_D of the ABAF to depict the influences among the sentences in $\mathcal{L}$ (Blümel et al. 2025; Rapberger, Ulbricht, and Wallner 2022). That is, for an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ the dependency graph $G_D = (V_D, E_D)$ is constructed as follows: we take a node $a \in V_D$ for each atom in $a \in \mathcal{L}$; for every rule $r \in \mathcal{R}$ we take:

- an edge $(a, b) \in E_D$ iff $head(r) = b$ and $a \in body(r)$,
- two symmetric edges (a, b) and (b, a) between each assumption node $a \in \mathcal{A}$ and its contrary node $b = \overline{a}$.

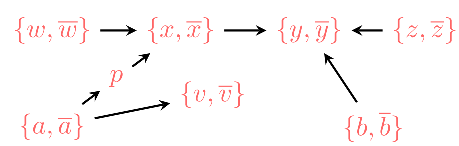
The first condition makes sure that influences among sentences encoded via rules are mirrored by the edges. Moreover, the second condition ensures that $S = atom(S)$ for any splitting set S when we later compute the SCCs.

Example 6. *Recall the ABAF D of Example 3. Its corresponding dependency graph $G_D = (V_D, E_D)$ is:*



We now have a simple directed graph, whose splittings correspond to the splittings of the ABAF, SETAF respectively. Thus, we can apply the strategy of Baumann, Brewka, and Wong (2011) to obtain a splitting set. In particular, after running Tarjan's algorithm on G_D to compute the SCCs, we contract every SCC to a single vertex leading to an acyclic graph G_D°, where nodes are the SCCs (highlighted in Example 6) of the original dependency graph.

Example 7. *We continue Example 6 and construct G_D°.*



As a result, each splitting set of D corresponds to exactly one splitting of G_D°. Further, Baumann, Brewka, and Wong (2011) focus on splitting that divides G_D° as evenly as possible in two parts with respect to the total number of nodes (Baumann, Brewka, and Wong 2011, Section 3.2, Algorithm 2). In this sense, a desirable splittings for G_D° in Example 7 could be $S = \{a, \overline{a}, b, \overline{b}, p, v, \overline{v}\}$.

Notice that the size of the dependency graph G_D is linear in the size of the ABAF (as $|V_D| = |\mathcal{L}|$) and the SCCs can be computed in linear time (Tarjan 1972). In addition, the present approach can also be employed to compute quasi-splittings. Once the dependency graph has been obtained, we contract only those pairs of nodes comprising an assumption and its contrary. Subsequently, we can apply the same techniques used for Dung-style AFs (Baumann et al. 2012), although these are more involved than those required for standard splitting. This strategy relies on the computation of minimum cuts in directed graphs using existing polynomial-time algorithms (Hao and Orlin 1994).

7 Relating ABA and SETAF Splittings

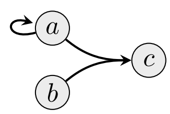
In this section, we compare splitting schemes for SETAFs and ABAFs. First, notice that for an ABAF D, every splitting set yields a splitting on the corresponding SETAF SF_D. Conversely, several ABA splitting sets may correspond to the same SETAF splitting.

Observation 1. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF and $SF_D = (A_D, R_D)$ the corresponding SETAF. Further, let S be a splitting set of D into D_1 and D_2. Then there is a splitting (SF_1, SF_2, R_3) of SF_D such that the assumptions in S are exactly the arguments of SF_1. Vice versa, for a SETAF SF and a splitting (SF_1, SF_2, R_3), there are several splitting sets $S_1, \ldots, S_n$ for the ABAF D_{SF} such that the arguments of SF_1 coincide with the assumptions in S_i.*

This opens the possibility of applying splitting on ABAFs before or after the instantiation into the corresponding SETAFs. To illustrate this point, consider the following.

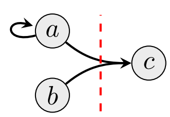
Example 8. *Consider the ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ (left) and its corresponding SETAF SF_D (right).*

$$\begin{aligned} \mathcal{A} &= \{a, b, c\} \\ \mathcal{L} &= \mathcal{A} \cup \overline{\mathcal{A}} \cup \{p, q, s\} \\ \mathcal{R} &= \{\overline{c} \leftarrow s, q, \;\; s \leftarrow b, \\ &\quad\;\; q \leftarrow p, \;\; p \leftarrow a, \;\; \overline{a} \leftarrow a\} \end{aligned}$$



The corresponding SETAF is $SF_D = (A_D, R_D)$ where $A_D = \mathcal{A}$ and $R_D = \{(a, a), (\{a, b\}, c)\}$. By appropriately combining $\{a, \overline{a}, b, \overline{b}\}$ with sentences in $\{p, q, s\}$ we get multiple splitting sets of D corresponding to a single SETAF splitting with $R_3 = \{(\{a, b\}, c)\}$.

$$\begin{aligned} S_1 &= \{a, b, \overline{a}, \overline{b}\} \\ S_2 &= S_1 \cup \{p\} \\ S_3 &= S_1 \cup \{s\} \\ S_4 &= S_1 \cup \{p, q, s\} \end{aligned}$$



Moreover, selecting $\{b\}$ as preferred extension of D_1 and SF_1, we add, respectively, the fresh assumption x_u attacking itself and c, and the self-attack (c, c) to SF_2. This results in $\emptyset$ being the only preferred extension of the $D_2^\star$ and $SF_2^\star$.

As the previous example showcases, both strategies allow for the same splittings and, indeed, compute the same expected result. When combining splitting with an instantiation based solver, the above considerations open different possibilities of how and when to split throughout the reasoning process. A first strategy, when instantiation is feasible, is to generate the SETAF and apply the splitting schema at the level of the graph. Alternatively, when this is not possible, one can first split the ABAF D into D_1 and D_2, then instantiate them into SETAFs SF_{D_1} and SF_{D_2} and apply the SETAF splitting algorithm, i.e. modify SF_{D_2} into $(SF_{D_2})^\star$ and compute extensions to be merged. This might allow one to utilise the SETAF splitting algorithm in cases where the argument graph of the entire ABAF is too hard to compute.

8 Related Work

We hereby clarify the relation of our splitting approach to SCC-recursiveness (Dvořák et al. 2024; Blümel et al. 2025), and we compare our notion of splitting to existing ones for logic programs (LPs) (Lifschitz and Turner 1994) and abstract dialectical frameworks (ADFs) (Linsbichler 2014), and syntax splitting (Parikh 1999).

In the incremental computation approach induced by the SCC-recursive property, one computes the extensions in subframeworks, and ultimately combines the thereby computed extension parts (as in the splitting approach). In contrast to splittings however, this is restricted to subframeworks that make up *strongly connected components* w.r.t. the primal graph of the SETAF or the ABAF's dependency graph (Dvořák et al. 2024; Blümel et al. 2025). Indeed, given the primal graph of a SETAF (resp. an ABAF's dependency graph), our approach corresponds to splitting between arguments (resp. sentences) such that the involved attacks (positive edges) have the same direction. However, splitting is more general in this regard, as the subframeworks do not have to be SCCs. Finally, SCC-recursiveness relies on a generalised semantics to deal with the decisions of prior parts of the framework, in contrast to the syntactic manipulation-based approach of splitting. This allows splitting to be implemented on top of existing ABA and SETAF solvers.

It has been shown that ABA captures normal logic programs with negation-as-failure under several semantics (Caminada and Schulz 2017). In particular, partial stable, well-founded and regular models in logic programming correspond to complete, grounded and preferred extensions in ABA. The translation works by turning every negative (resp. positive) literals of an LP into an assumption (resp. contrary), and vice versa. By means of this simple transformation, our splitting algorithm can be adapted in the context of logic programming to obtain a divide-and-conquer approach beyond stable semantics, thus generalising the original result (Lifschitz and Turner 1994). As a by-product, we are then able to split normal logic programs under semantics based on partially stable, well-founded and regular models.

ADFs (Brewka et al. 2018) are an expressive argumentation formalism, where each argument is associated with a propositional formula over arguments as variables as an acceptance condition. It is well-known that SETAFs can be interpreted as a special kind of ADFs with acceptance conditions in the form of a conjunction of disjunctive clauses of negated literals (Dvořák, Keshavarzi Zafarghandi, and Woltran 2023). That is, in principle we can apply ADF splitting to SETAFs. However, it is not clear that following the ADF approach the modified second framework again is of the desired (SETAF-like) form and whether one can avoid certain overheads in the simpler case of SETAFs. Upon closer inspection and with minor syntactic manipulation the ADF approach in the special case of SETAF-like frameworks is similar to introducing an artificial self-attacking argument that behaves similarly to Definition 14. However, such a trick is not needed for SETAFs in our case.

Parikh (1999) introduced the concept of *syntax splitting* to characterize belief sets that contain independent pieces of information. In this setting, a syntax splitting consists of a partition of the knowledge base into independent components, where each independent piece of information can be represented using only one part of these partition. In this paper, we have considered a notion of splitting which is more general, where the different parts of the knowledge base may not be independent from each other. For this, we provided suitable modifications to account for such interdependencies.

9 Conclusion and Future Work

In this paper, we have presented a modification-based approach to splitting assumption-based argumentation frameworks. First, we have introduced a splitting schema for SETAFs, to make splitting available on ABAFs after instantiation. Leveraging on the close connection between SETAFs and ABAFs, we have thus proposed a way to split when reasoning in ABA is performed indirectly via the corresponding argument graph. Moreover, to overcome the requirement of an instantiation and its associated costs, we have introduced a splitting schema that works directly on ABA knowledge bases. For both approaches, we have shown that extensions of a given ABAF can be obtained incrementally from its sub-frameworks, by means of simple syntactic modifications. Conversely, we can project an arbitrary extension of the whole framework to its sub-frameworks. Since this is bound to the specific structure of the underlying ABAF, we have considered a more general variant of splitting called parametrised splitting inspired by Baumann et al. (2012). Moreover, it is easy to see that each of the steps involved can be carried out efficiently and implemented on top of common ABA (or SETAF) solvers. Indeed, the splitting techniques introduced in the paper require only syntactic modifications to the sub-frameworks, which do not lead to an exponential increase in size (and may, in some cases, even reduce it). In particular, through the reduct and modification, we introduce at most one additional rule and one additional assumption with respect to the original sub-framework. In the case of parametrised splitting, the number of additional rules and assumptions is bounded by $2k$ for a k-splitting. Therefore, an obvious next step is to implement our algorithm and perform an experimental evaluation in the spirit of Baumann, Brewka, and Wong (2011), where executions with splitting result in an average acceleration of 60% in comparison to executions without splitting. In particular, we believe that parametrised splitting could be helpful in the context of the recently proposed Argumentative Causal Discovery (Russo, Rapberger, and Toni 2024). In fact, this framework faces a major challenge in terms of its scalability, as it exhibits suboptimal performance on larger instances.

AI Declaration

The authors have not employed any Generative AI tools.

Acknowledgments

This paper is an extended and revised version of two workshop papers that were presented in SAFA 2024 (Buraglio et al. 2024a) and NMR 2025 (Buraglio 2025). We are grateful to the anonymous reviewers for their valuable comments and suggestions on a preliminary version of this paper. Moreover, the authors would like to thank Matthias König for his contributions to the paper presented at SAFA 2024. This work has been supported by the Austrian Science Fund (FWF) under grant 10.55776/COE122 and by the European Union's Horizon 2020 research and innovation programme (under grant agreement 101034440).

References

Baroni, P.; Giacomin, M.; and Liao, B. 2014. On topology-related properties of abstract argumentation semantics. A correction and extension to dynamics of argumentation systems: A division-based method. *Artif. Intell.* 212:104–115.

Baumann, R.; Brewka, G.; Dvořák, W.; and Woltran, S. 2012. Parameterized splitting: A simple modification-based approach. In Erdem, E.; Lee, J.; Lierler, Y.; and Pearce, D., eds., *Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz*, volume 7265 of *Lecture Notes in Computer Science*, 57–71. Springer.

Baumann, R.; Brewka, G.; and Wong, R. 2011. Splitting argumentation frameworks: An empirical evaluation. In Modgil, S.; Oren, N.; and Toni, F., eds., *Theorie and Applications of Formal Argumentation - First International Workshop, TAFA 2011. Barcelona, Spain, July 16-17, 2011, Revised Selected Papers*, volume 7132 of *Lecture Notes in Computer Science*, 17–31. Springer.

Baumann, R. 2011. Splitting an argumentation framework. In Delgrande, J. P., and Faber, W., eds., *Logic Programming and Nonmonotonic Reasoning - 11th International Conference, LPNMR 2011, Vancouver, Canada, May 16-19, 2011. Proceedings*, volume 6645 of *Lecture Notes in Computer Science*, 40–53. Springer.

Berthold, M.; Rapberger, A.; and Ulbricht, M. 2024. Capturing non-flat assumption-based argumentation with bipolar setafs. In Marquis, P.; Ortiz, M.; and Pagnucco, M., eds., *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024*.

Blümel, L.; Rapberger, A.; Thimm, M.; and Toni, F. 2025. On independence and scc-recursiveness in assumption-based argumentation. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 4382–4390. International Joint Conferences on Artificial Intelligence Organization.

Bondarenko, A.; Dung, P. M.; Kowalski, R. A.; and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.* 93:63–101.

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J. P.; and Woltran, S. 2018. Abstract dialectical frameworks. In *Handbook of Formal Argumentation*. College Publications. chapter 5, 237–285. also appears in IfCoLog Journal of Logics and their Applications 4(8):2263–2318.

Buraglio, G.; Dvorák, W.; König, M.; and Woltran, S. 2024a. Splitting argumentation frameworks with collective attacks. In Borg, A.; Ellmauthaler, S.; Mailly, J.; and Niskanen, A., eds., *Proceedings of the Fifth International Work-*

shop on Systems and Algorithms for Formal Argumentation co-located with 10th International Conference on Computational Models of Argument (COMMA 2024), Hagen, Germany, September 17th, 2024, CEUR Workshop Proceedings, 41–55. CEUR-WS.org.

Buraglio, G.; Dvořák, W.; König, M.; and Ulbricht, M. 2024b. Justifying argument acceptance with collective attacks: Discussions and disputes. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 3281–3288. ijcai.org.

Buraglio, G.; Dvorak, W.; and Woltran, S. 2026. Splitting assumption-based argumentation frameworks. CoRR abs/2604.27964.

Buraglio, G. 2025. Splitting assumption-based argumentation frameworks. In Rapberger, A., and Rudolph, S., eds., *Proceedings of the 23rd International Workshop on Non-Monotonic Reasoning (NMR 2025) co-located with the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR 2025), Melbourne, Australia, November 11-13, 2025*, CEUR Workshop Proceedings, 17–31. CEUR-WS.org.

Caminada, M., and Schulz, C. 2017. On the equivalence between assumption-based argumentation and logic programming. *J. Artif. Intell. Res.* 60:779–825.

Carrera, Á., and Iglesias, C. A. 2015. A systematic review of argumentation techniques for multi-agent systems research. *Artif. Intell. Rev.* 44(4):509–535.

Cyras, K.; Fan, X.; Schulz, C.; and Toni, F. 2018. Assumption-based argumentation: Disputes, explanations, preferences. In *Handbook of Formal Argumentation*. College Publications. chapter 7, 365–408.

Cyras, K.; Heinrich, Q.; and Toni, F. 2021. Computational complexity of flat and generic assumption-based argumentation, with and without probabilities. *Artif. Intell.* 293:103449.

Dimopoulos, Y.; Dvořák, W.; König, M.; Rapberger, A.; Ulbricht, M.; and Woltran, S. 2024. Redefining ABA+ semantics via abstract set-to-set attacks. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 10493–10500. AAAI Press.

Dimopoulos, Y.; Mailly, J.; and Moraitis, P. 2019. Argumentation-based negotiation with incomplete opponent profiles. In Elkind, E.; Veloso, M.; Agmon, N.; and Taylor, M. E., eds., *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, 1252–1260. International Foundation for Autonomous Agents and Multiagent Systems.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–358.

Dvořák, W.; König, M.; Ulbricht, M.; and Woltran, S. 2024. Principles and their computational consequences for argumentation frameworks with collective attacks. *J. Artif. Intell. Res.* 79:69–136.

Dvořák, W.; Greßler, A.; and Woltran, S. 2018. Evaluating setafs via answer-set programming. In Thimm, M.; Cerutti, F.; and Vallati, M., eds., *Proceedings of the Second International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2018) co-located with the 7th International Conference on Computational Models of Argument (COMMA 2018), Warsaw, Poland, September 11, 2018*, volume 2171 of *CEUR Workshop Proceedings*, 10–21. CEUR-WS.org.

Dvořák, W.; Keshavarzi Zafarghandi, A.; and Woltran, S. 2023. Expressiveness of SETAFs and support-free ADFs under 3-valued semantics. *J. Appl. Non Class. Logics* 33(3-4):298–327.

Fan, X., and Toni, F. 2012. Agent strategies for aba-based information-seeking and inquiry dialogues. In Raedt, L. D.; Bessiere, C.; Dubois, D.; Doherty, P.; Frasconi, P.; Heintz, F.; and Lucas, P. J. F., eds., *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, 324–329. IOS Press.

Fan, X.; Toni, F.; Mocanu, A.; and Williams, M. 2014. Dialogical two-agent decision making with assumption-based argumentation. In *AAMAS '14*, 533–540.

Fan, X. 2018. On generating explainable plans with assumption-based argumentation. In Miller, T.; Oren, N.; Sakurai, Y.; Noda, I.; Savarimuthu, B. T. R.; and Son, T. C., eds., *PRIMA 2018: Principles and Practice of Multi-Agent Systems - 21st International Conference, Tokyo, Japan, October 29 - November 2, 2018, Proceedings*, volume 11224 of *Lecture Notes in Computer Science*, 344–361. Springer.

Gabbay, D.; Giacomin, M.; Simari, G. R.; and Thimm, M., eds. 2021. *Handbook of Formal Argumentation*, volume 2. College Publications.

Gao, Y.; Toni, F.; Wang, H.; and Xu, F. 2016. Argumentation-based multi-agent decision making with privacy preserved. In Jonker, C. M.; Marsella, S.; Thangarajah, J.; and Tuyls, K., eds., *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, 1153–1161. ACM.

Greßler, A.; Dvořák, W.; and Woltran, S. 2024. The GSAF solver and verifier. In Reed, C.; Thimm, M.; and Rienstra, T., eds., *Computational Models of Argument - Proceedings of COMMA 2024, Hagen, Germany, September 18-20, 2024*, volume 388 of *Frontiers in Artificial Intelligence and Applications*, 353–354. IOS Press.

Hadoux, E.; Hunter, A.; and Polberg, S. 2023. Strategic argumentation dialogues for persuasion: Framework and experiments based on modelling the beliefs and concerns of the persuadee. *Argument Comput.* 14(2):109–161.

Hao, J., and Orlin, J. 1994. A faster algorithm for finding the minimum cut in a directed graph. *Journal of Algorithms* 17(3):424–446.

König, M.; Rapberger, A.; and Ulbricht, M. 2022. Just a matter of perspective. In Toni, F.; Polberg, S.; Booth, R.; Caminada, M.; and Kido, H., eds., *Computational Models of Argument - Proceedings of COMMA 2022, Cardiff, Wales, UK, 14-16 September 2022*, volume 353 of *Frontiers in Artificial Intelligence and Applications*, 212–223. IOS Press.

Lehtonen, T.; Rapberger, A.; Ulbricht, M.; and Wallner, J. P. 2023. Argumentation frameworks induced by assumption-based argumentation: Relating size and complexity. In Marquis, P.; Son, T. C.; and Kern-Isberner, G., eds., *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023*, 440–450.

Lehtonen, T.; Rapberger, A.; Toni, F.; Ulbricht, M.; and Wallner, J. P. 2024. Instantiations and computational aspects of non-flat assumption-based argumentation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 3457–3465. ijcai.org.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2021a. Declarative algorithms and complexity results for assumption-based argumentation. *J. Artif. Intell. Res.* 71:265–318.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2021b. Harnessing incremental answer set solving for reasoning in assumption-based argumentation. *Theory Pract. Log. Program.* 21(6):717–734.

Liao, B. 2013. Toward incremental computation of argumentation semantics: A decomposition-based approach. *Ann. Math. Artif. Intell.* 67(3-4):319–358.

Lifschitz, V., and Turner, H. 1994. Splitting a logic program. In Hentenryck, P. V., ed., *Logic Programming, Proceedings of the Eleventh International Conference on Logic Programming, Santa Marherita Ligure, Italy, June 13-18, 1994*, 23–37. MIT Press.

Linsbichler, T. 2014. Splitting abstract dialectical frameworks. In Parsons, S.; Oren, N.; Reed, C.; and Cerutti, F., eds., *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, 357–368. IOS Press.

Nielsen, S. H., and Parsons, S. 2006. A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In Maudet, N.; Parsons, S.; and Rahwan, I., eds., *Argumentation in Multi-Agent Systems, Third International Workshop, ArgMAS 2006, Hakodate, Japan, May 8, 2006, Revised Selected and Invited Papers*, volume 4766 of *Lecture Notes in Computer Science*, 54–73. Springer.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. *Logic, Language and Computation* 2:266–278.

Rapberger, A.; Ulbricht, M.; and Wallner, J. P. 2022. Argumentation frameworks induced by assumption-based argumentation: Relating size and complexity. In Arieli, O.; Casini, G.; and Giordano, L., eds., *Proceedings of the 20th International Workshop on Non-Monotonic Reasoning, NMR 2022, Part of the Federated Logic Conference (FLoC 2022), Haifa, Israel, August 7-9, 2022*, volume 3197 of *CEUR Workshop Proceedings*, 92–103. CEUR-WS.org.

Russo, F.; Rapberger, A.; and Toni, F. 2024. Argumentative causal discovery. In Marquis, P.; Ortiz, M.; and Pagnucco, M., eds., *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024*.

Tarjan, R. E. 1972. Depth-first search and linear graph algorithms. *SIAM J. Comput.* 1(2):146–160.

Toni, F. 2013. A generalised framework for dispute derivations in assumption-based argumentation. *Artif. Intell.* 195:1–43.

Turner, H. 1996. Splitting a default theory. In Clancey, W. J., and Weld, D. S., eds., *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, USA, August 4-8, 1996, Volume 1*, 645–651. AAAI Press / The MIT Press.