

# Splitting Argumentation Frameworks with Collective Attacks

Giovanni Buraglio<sup>1</sup>, Wolfgang Dvořák<sup>1</sup>, Matthias König<sup>1</sup> and Stefan Woltran<sup>1</sup>

<sup>1</sup>TU Wien, Institute of Logic and Computation

## Abstract

A recurring notion in the abstract argumentation community is that of collective attacks, whereby a set of arguments, rather than a single one, attacks another argument. The resulting frameworks capturing this phenomenon are referred to as SETAFs. Given the possibility of facing an exponential runtime in the size of the framework, techniques have been presented to compute extensions of a given framework incrementally, i.e. restricting the search space to sub-frameworks only, and then combining the obtained results. Existing research has primarily focused on approaches based on SCC-recursiveness, where SETAFs are evaluated along their strongly connected components (SCCs) using generalized semantics and dedicated algorithms. Splitting approaches are more general in this regard, as they do not have to consider SCCs individually and can be used on top of arbitrary argumentation solvers. Splitting techniques have been successfully applied in abstract argumentation but have been neglected for SETAFs so far. Towards filling this gap our work investigates the concept of (modification-based) splitting for SETAFs. We show that a splitting-based approach is possible for common semantics (such as admissible, complete, grounded, preferred, and stable), generalizing corresponding results of AFs, which can be seen as a restricted class of SETAFs. Along the way, we point out intricate details that are obvious or trivial for AFs, but help us to understand the underlying ideas in greater detail than before.

## Keywords

collective attacks, splitting, abstract argumentation

## 1. Introduction

Most argumentation problems are intractable in general, which means even the best-known methods for solving these problems oftentimes cannot avoid exploring the whole exponential size solution-space. The fewer arguments we have to consider for each problem, the more efficient an approach can be in general, which is why in the setting of abstract argumentation (see [1]) it can be advantageous to evaluate only *parts* of a framework at once and then combine the results. One approach to do this is *splitting*, as introduced by Baumann for Dung-style argumentation frameworks (AFs) [2] and later considered in other settings [3, 4, 5]. A popular syntactic addition within argumentation frameworks that has been regarded in the community

*SAFA'24: Fifth International Workshop on Systems and Algorithms for Formal Argumentation, September 17, 2024, Hagen, Germany*

✉ giovanni.buraglio@tuwien.ac.at (G. Buraglio); wolfgang.dvorak@tuwien.ac.at (W. Dvořák);

matthias.koenig@tuwien.ac.at (M. König); stefan.woltran@tuwien.ac.at (S. Woltran)

🌐 <https://informatics.tuwien.ac.at/people/giovanni-buraglio> (G. Buraglio); <https://dbai.tuwien.ac.at/staff/dvorak/>

(W. Dvořák); <https://dbai.tuwien.ac.at/user/mkoenig/> (M. König); <https://www.dbai.tuwien.ac.at/staff/woltran/>

(S. Woltran)

📄 0009-0004-9592-4739 (G. Buraglio); 0000-0002-2269-8193 (W. Dvořák); 0000-0003-0205-0039 (M. König);

0000-0003-1594-8972 (S. Woltran)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

are collective attacks (due to Nielsen and Parsons [6]), the resulting frameworks are called SETAFs [7, 8, 9]. Collective attacks have proven useful for the instantiation of structured argumentation, see e.g. [10, 11, 12]. However, a splitting approach for SETAFs has not yet been introduced; in this paper we close this gap and investigate interesting differences and similarities to the simpler AF-case, which ultimately shines a new light on the existing ideas on splitting. We show that if we carefully generalize the underlying intuitions, SETAFs yield an elegant splitting procedure for most common semantics. Splitting techniques have also been introduced for Abstract Dialectical Frameworks (ADFs) [3]. However, even though SETAFs can be modeled through ADFs, it is not obvious how these results are applicable in the context of SETAFs—as ADFs deal with simple links and propositional acceptance conditions whereas SETAFs rely on directed hypergraphs.

Similar to the structure of Baumann [2], on this work, we consider *directional splitting*, which means that we split a large SETAF  $SF$  into two sub-frameworks  $SF_1$  and  $SF_2$  in a way s.t. only  $SF_1$  influences  $SF_2$ , but not the other way around. Clearly, the choices within the subframework  $SF_1$  determine the acceptable arguments in  $SF_2$ , which has to be accounted for in order to correctly solve reasoning problems. In general, for such a scenario there are two approaches: (a)  $SF_2$  is evaluated w.r.t. generalized semantics, that take the decisions in  $SF_1$  into account, or (b) the modification-based approach, i.e.,  $SF_2$  is syntactically modified, to simulate the effects of the decisions in  $SF_1$ . While (a) is used for example in SCC-recursiveness [8] (i.e., an approach based on the strongly-connected components), AFs with input/output [13], or decomposition-based techniques for AFs [14], approach (b) allows us to use unmodified out-of-the-box argumentation solvers on both sub-frameworks. In this paper we will present a modification-based splitting approach for SETAFs. This paper is organized as follows.

- In Section 2 we recall the definition of SETAFs and their semantics, as well as the splitting approach in AFs.
- Throughout Section 3 we explain the intuitions and intricacies of splitting in the context of collective attacks and justify our design choices for the splitting algorithm.
- Section 4 contains the resulting definition of the splitting approach for SETAFs, as well as the theoretical underpinning for its correctness. Moreover, we establish its connection to the *directionality* principle—a desirable result we already know from AFs.
- Finally, in Section 5 we conclude and discuss related work.

## 2. Background

In this section, we recall the definition of argumentation frameworks with collective attacks (SETAFs) [6, 7, 9] and their semantics. As we will see, with a slight abuse of notation, we can view Dung-style argumentation frameworks (AFs) [1] as a special case of SETAFs.

**Definition 1.** A SETAF is a pair  $SF = (A, R)$  where  $A$  is a finite set of arguments, and  $R \subseteq (2^A \setminus \{\emptyset\}) \times A$  is the attack relation. For an attack  $(T, h) \in R$  we call  $T$  the tail and  $h$  the head of the attack. SETAFs  $(A, R)$ , where for all  $(T, h) \in R$  it holds that  $|T| = 1$ , amount to (standard Dung) AFs. We usually 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  $S_R^+$  to denote the set  $\{a \mid S \text{ attacks } a\}$  and define the range of  $S$  (w.r.t.  $R$ ), denoted  $S_R^\oplus$ , as the set  $S \cup S_R^+$ .

The fundamental notions of conflict and defense from Dung-style AFs naturally generalize to SETAFs. These notions are the basis for the semantics we investigate in this paper.

**Definition 2.** Let  $SF = (A, R)$  be a SETAF. A set  $S \subseteq A$  is conflicting in  $SF$  if  $S$  attacks  $a$  for some  $a \in S$ .  $S \subseteq A$  is conflict-free in  $SF$ , if  $S$  is not conflicting in  $SF$ , i.e. if  $T \cup \{h\} \not\subseteq S$  for each  $(T, h) \in R$ .  $cf(SF)$  denotes the set of all conflict-free sets in  $SF$ . An argument  $a \in A$  is defended (in  $SF$ ) by a set  $S \subseteq A$  if for each  $B \subseteq A$ , such that  $B$  attacks  $a$ , also  $S$  attacks  $B$  in  $SF$ . A set  $T \subseteq A$  is defended (in  $SF$ ) by  $S$  if each  $a \in T$  is defended by  $S$  (in  $SF$ ).

The semantics we study in this work are the admissible, complete, grounded, preferred, and stable semantics, which we will abbreviate by *adm*, *com*, *grd*, *pref*, and *stb*, respectively [6, 7, 15]. Moreover, we use  $\sigma(SF)$  to denote the set of extensions of  $SF$  under semantics  $\sigma$ .

**Definition 3.** Given a SETAF  $SF = (A, R)$  and a conflict-free set  $S \in cf(SF)$ . Then,

- $S \in adm(SF)$ , if  $S$  defends itself in  $SF$ ,
- $S \in com(SF)$ , if  $S \in adm(SF)$  and  $a \in S$  for all  $a \in A$  defended by  $S$ ,
- $S \in grd(SF)$ , if  $S \in com(SF)$  and there is no  $T \in com(SF)$  s.t.  $T \subset S$ ,
- $S \in pref(SF)$ , if  $S \in adm(SF)$  and there is no  $T \in adm(SF)$  s.t.  $T \supset S$ , and
- $S \in stb(SF)$ , if  $S_R^\oplus = A$ .

The relationship between the semantics has been clarified in [6, 7, 15] and matches with the relations between the semantics for Dung AFs, i.e. for any SETAF  $SF$ :

$$stb(SF) \subseteq pref(SF) \subseteq com(SF) \subseteq adm(SF) \subseteq cf(SF)$$

We now recall Baumann's splitting approach for AFs [2] (in a slightly adapted equivalent form to fit our notation).

**Definition 4.** Let  $F = (A, R)$  be an AF,  $F_1 = (A_1, R_1)$  and  $F_2 = (A_2, R_2)$  two sub-frameworks of  $SF$  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  $R_3 \subseteq A_1 \times A_2$ . We call the triple  $(F_1, F_2, R_3)$  a splitting of  $F$ . For such a splitting the  $(E, R_3)$ -reduct w.r.t.  $E \subseteq A_1$  is the AF  $AF' = (A', R')$  with  $A' = A_2 \setminus E_{R_3}^+$  and  $R' = R_2 \cap (A' \times A')$ . The set of undecided arguments w.r.t.  $E \subseteq A_1$  is defined as  $U_E = A_1 \setminus E_{R_1}^\oplus$ .

We will later generalize the notion of the reduct to be applicable in the context of SETAFs.

**Definition 5.** 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 defined as  $mod_{U_E, R_3}(F'_2) = (A'_2, R'_2 \cup \{(b, b) \mid \exists a \in U_E : (a, b) \in R_3\})$ .

It is easy to see that the definition of the modification does not actually rely on the undecided arguments, but rather uses the arguments as means to obtain the links which stem from undecided arguments. Later on, we will make use of this fact to simplify the respective notions. Baumann [2] showed that by this definition it is possible for a splitting to compute the extensions for each sub-framework separately.

**Theorem 6** ([2]). *Let  $(F_1, F_2, R_3)$  be a splitting for an AF  $F = (A, R)$  with  $F_i = (A_i, R_i)$  and  $\sigma \in \{cf, adm, stb, com, pref, grd\}$ .*

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  $E \cap A_2 \in \sigma(mod_{U_{E,R_3}}(F_2'))$ .*

### 3. Towards Splitting for SETAFs

In this section, we introduce fundamental ideas for defining divide and conquer algorithms based on splitting in the presence of collective attacks. As a starting point, we generalize the notion of splitting introduced by Baumann [2] for Dung-style AFs. For this, we will in the following provide comprehensive intuitions.

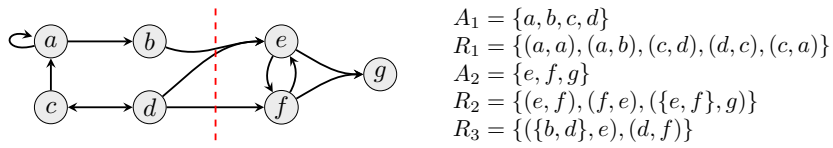
**Definition 7.** *Let  $SF = (A, R)$  be a SETAF,  $SF_1 = (A_1, R_1)$  and  $SF_2 = (A_2, R_2)$  two sub-frameworks of  $SF$  such that  $A_1 \cap A_2 = \emptyset$ ,  $A = A_1 \cup A_2$  and  $R = R_1 \cup R_2 \cup R_3$  with  $R_3 \subseteq ((2^{A_1} \setminus \{\emptyset\}) \cup 2^{A_2}) \times A_2$ . We call a splitting of  $SF$  the triple  $(SF_1, SF_2, R_3)$ . Moreover, we say that  $R_3$  is the set of links of the splitting  $(SF_1, SF_2, R_3)$ .*

In a nutshell, we investigate a large SETAF  $SF$  that has two sub-frameworks  $SF_1$  and  $SF_2$  with attacks within themselves, and attacks  $R_3$  that stem from  $A_1$  (at least in part) and target only arguments in  $SF_2$ . The general idea is to compute extensions of  $SF_1$  and  $SF_2$ , which combined give us extensions of  $SF$ . 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.

#### 3.1. Simple Splitting for SETAFs

We take here into account SETAF splittings where the *whole* tail of the links is separated from the respective heads. Notice that, whereas these are indeed straightforward generalizations of AF splittings [2], they represent only a specific class of the splittings introduced by Definition 7. In Section 3.2, we will investigate the problem of splitting SETAFs in full generality.

**Example 8.** *Consider the SETAF  $SF$  below with its splitting  $(SF_1, SF_2, R_3)$ , with  $SF_1 = (A_1, R_1)$  and  $SF_2 = (A_2, R_2)$ . The dashed line indicates the separation between the sub-frameworks. The goal is to compute the extensions of  $SF$  by computing the extensions of  $SF_1$  and (a modified version of)  $SF_2$  separately. Note that throughout the remaining part of this paper we will refer to the splitting in question always as  $(SF_1, SF_2, R_3)$ , unless indicated otherwise.*



One can verify that  $SF_1$  has two preferred extensions:  $E_1 = \{b, c\}$  and  $E_2 = \{d\}$ . Let us first consider  $E_1$ . Since  $d$  is defeated in  $SF_1$  we can argue that the links (i.e., attacks in  $R_3$ ) do not at all affect the arguments in  $SF_2$ , and we can just evaluate  $SF_2$  “as is” to obtain the extensions

$E_{1,1} = \{e, g\}$  and  $E_{1,2} = \{f, g\}$ . We can combine the extensions from  $SF_1$  and  $SF_2$  to obtain  $\{b, c, e, g\}$  and  $\{b, c, f, g\}$ , which are indeed preferred extensions of  $SF$ . Now on the other hand if we consider  $E_2$ , it is non-trivial how this affects  $SF_2$ : while  $f$  is defeated by  $d$ ,  $e$  is targeted by an attack from the accepted argument  $d$  and the argument  $b$ , which in  $SF_1$  is neither accepted nor outright rejected by  $E_2$ . In the following, we will argue how to properly deal with these cases and introduce a splitting method that correctly characterizes all extensions.

In the example above we can see that the status of the arguments in  $SF_1$  w.r.t. the extension of  $SF_1$  we investigate determines whether and how the arguments in  $SF_2$  are affected. Indeed, in the AF case the status of the single argument in the tail of a link solely determines whether the head is removed or not from the second sub-framework, or whether a self-attack is added to it (c.f. Definitions 4 and 5). Similarly for SETAFs, it is possible to distinguish three different scenarios for the status of the arguments in  $A_1$  after evaluating  $SF_1$ , corresponding to the cases where the argument is accepted (i.e., in an extension  $E_1 \in \sigma(SF_1)$ ), defeated (in  $(E_1)_{R_1}^+$ ) or undecided (in  $A_1 \setminus (E_1)_{R_1}^\oplus$ ). Note however, that while on AFs the status of a link and its one tail argument coincide, for SETAFs links (like any other attack) can have *multiple* tail arguments. Hence, the status of a link  $(T, h) \in R_3$  of a splitting  $(SF_1, SF_2, R_3)$  can be determined after evaluating  $SF_1$  as follows: (i) *all* of the arguments in the tail of a link (that are also in  $SF_1$ ) are accepted (i.e., in an extension  $E_1 \in \sigma(SF_1)$ ), (ii) at least one argument in the tail of a link is defeated by  $E_1$  (in  $(E_1)_{R_1 \cup R_3}^+$ ) or (iii) no argument is defeated but at least one is undecided (in  $A_1 \setminus (E_1)_{R_1}^\oplus$ ). In what follows, we consider cases (i)-(iii) separately and show an intuition on how these need to be treated. We start with (i) in the Example 9, depicting a situation where the attack  $(\{x, y\}, z)$  is *in*. As one can see, in certain circumstances the above definition of a splitting corresponds to a straightforward generalization of the AF case.

**Example 9.** Consider the AF  $F$  (left), and the SETAF  $SF$  (right) with its splitting  $(SF_1, SF_2, R_3)$ .



We look at the preferred extensions  $\{a\}$  for the first part of  $F$ , and see that  $b$  in the second part is defeated. Thus, by the approach of [2],  $b$  is removed when we look at the modified second framework, and its outgoing attack towards  $c$  as well. We trivially obtain  $\{c\}$  as a preferred extension for the modified right part. We combine  $\{a\} \cup \{c\}$  to obtain  $\{a, c\}$  as the only preferred extension for  $F$ .

Analogously, we have  $\{v, w\}$  as a preferred extension in  $SF_1$ . Hence, we remove  $x$  from  $SF_2$  to obtain the modified framework  $SF_2^*$ . As in the AF case, any outgoing attack from  $x$  (no matter if other arguments are in the tail, like  $y$  in our case) cannot affect their targeted argument ( $z$ ), as  $z$  is defended against this attack. Hence, we remove the entire attack and obtain  $SF_2^* = (\{y, z\}, \emptyset)$  which trivially yields  $\{y, z\}$  as its preferred extension. As a result, one gets  $\{v, w\} \cup \{y, z\}$  as a preferred extension for  $SF$ .

We next discuss case (ii) for the status of a link in Example 10 below.

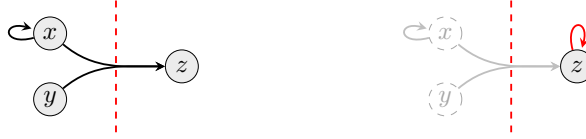
**Example 10.** For the SETAF  $SF$  below, we identify the preferred extension  $\{w, y\}$  in the first part (i.e.,  $\{w, y\} \in \text{pref}(SF_1)$ ). Given that  $w \in E$ , we get that  $x$  is defeated, which means the link

$(\{x, y\}, z)$  can be seen as out. Intuitively,  $z$  needs no more counter-attack for this incoming attack—since  $x$  is already defeated. For this reason, such an attack does not affect the modification of  $SF_2$ . Hence, the modified  $SF_2^*$  is as depicted on the right, preserving  $z$  as an acceptable argument. Note also that any possible outgoing attacks of  $z$  remain untouched. We thus have  $\{w, y\} \in \text{pref}(SF_1)$  and  $\{z\} \in \text{pref}(SF_2^*)$  as preferred extensions of  $SF_1$  and  $SF_2^*$ , respectively. It is easy to see that  $\{w, y\} \cup \{z\}$  is a preferred extension of  $SF$ .



Different considerations are due whenever the original SETAF contains a link which is *undecided*, corresponding to case (iii).

**Example 11.** Consider the SETAF  $SF$  displayed below (left) and  $E = \{y\} \in \text{pref}(SF_1)$ .  $x$  is undecided w.r.t.  $E$ , i.e.,  $x \in A_1 \setminus E_{R_1}^\oplus$ . This makes the status of  $(\{x, y\}, z)$  undecided as well, enforcing a modification of the right part of the SETAF. Analogous to the approach for AFs, we add a self-attack on  $z$ , obtaining  $SF_2^* = (\{z\}, \{(z, z)\})$  (right). Intuitively, this models the fact that  $z$  cannot be accepted in  $SF_2^*$ , since there is at least one attack that  $z$  is not defended against. On the other hand,  $z$  is not rejected, i.e., if  $z$  were to attack other arguments in  $SF_2$  they need defence against  $z$ . Hence, we cannot outright remove  $z$  as in case (ii). In this situation, one gets  $\emptyset \in \text{pref}(SF_2^*)$ . Hence,  $\{y\} \cup \emptyset$  is a preferred extension of  $SF$ .



Examples 9, 10, and 11 above display only a special case of splitting for SETAFs, where the *whole* tail of an attack is separated from its target. Hence, as of now performing splitting-based techniques in the presence of collective attacks represents an easy and straightforward generalization of its AF counterpart. This stems from the fact that modification is only slightly impacted by the presence of multiple arguments in the tail of a link. Consequently, minor adjustments are needed to handle such situations. However, due to their rich syntax, SETAFs allow for another possible way to separate an attack via splitting. In the following, we consider SETAF splittings in full generality.

### 3.2. Diagonal Splitting for SETAFs

The enriched syntax of SETAFs allows us to take into account splittings that separate arguments taking part in the same collective attack. In particular, it is possible to split a SETAF in such a way that two parts of the same tail of a link end up being in different sub-frameworks. This is captured by the possibility of having  $R_3 \subseteq ((2^{A_1} \setminus \{\emptyset\}) \cup 2^{A_2}) \times A_2$  as for our Definition 7. We investigate such scenarios in connection with the cases (i)-(iii) as before. Note however that we need to consider the cases more carefully, as we now also consider links  $(T, h) \in R_3$  where the tail  $T$  is spread over both  $A_1$  and  $A_2$ . More formally we call a link  $(T, h) \in R_3$ :

- (i) in iff  $\nexists a \in T$  s.t.  $a \in E_{R_1 \cup R_3}^+$  and  $\forall a \in T \cap A_1, a \in E$ ,
- (ii) out iff  $\exists a \in T$  s.t.  $a \in E_{R_1 \cup R_3}^+$  or
- (iii) undec iff  $\nexists a \in T$  s.t.  $a \in E_{R_1 \cup R_3}^+$  and  $\exists a \in T \cap A_1$  s.t.  $a \in A \setminus E_{R_1}^\oplus$ .

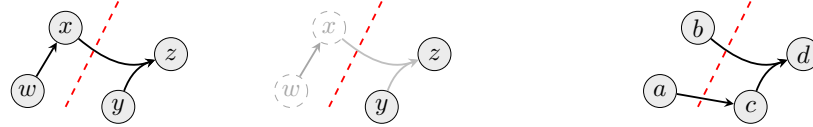
We again give the necessary intuitions for each case, starting with (i).

**Example 12.** For the below SETAF  $SF$  (left), we consider  $\{x\} \in \text{adm}(SF_1)$ . Therefore, intuitively it is the remaining part of the attack stemming from  $y$  that is decisive for the status of the target argument  $z$ . Since  $x$  is accepted, it suffices for the success of the attack  $(\{x, y\}, z)$  to consider the status of  $y$  alone. Therefore, one can solely consider the remaining part of the attack in  $SF_2$ . Resulting from this, we obtain  $SF_2^* = (\{y, z\}, \{(y, z)\})$  (right). Given that  $\{y\} \in \text{adm}(SF_2^*)$ , we retrieve  $\{x\} \cup \{y\}$  as an admissible set of  $SF$ .



More generally, given a set of arguments  $E$  which is accepted in  $SF_1$ , the success of any link  $(T, h) \in R_3$  where  $T \cap A_1 \subseteq E$  is dependent only on the status of  $T \setminus E$  in  $SF_2$ . Opposite considerations can be made when arguments in the links' tails are defeated (case (ii)).

**Example 13.** Consider the following SETAF  $SF$  (left) with  $E = \{w\} \in \text{adm}(SF_1)$ . As in Example 10, the argument  $w \in E$  attacks part of the link's tail (i.e.  $x$ ), thereby neutralizing the collective attack  $(\{x, y\}, z)$ . As a result,  $w$  is compatible with both  $y$  and  $z$ . More formally, the rightmost part of  $SF$  is modified to obtain  $SF_2^* = (\{y, z\}, \emptyset)$  (middle). Indeed, their set-union  $\{w\} \cup \{y, z\}$  is an admissible set of  $SF$ . Note that this case (ii) can also occur via a link. Consider the splitting for the SETAF  $SF^\dagger$  (right) where  $\{a, b\}$  is a preferred extension in the left sub-framework. Since  $a$  defeats  $c$ , the link  $(\{b, c\}, d)$  has to be deleted in the modified framework, even though all tail-arguments of the link within  $A_1$  (i.e., in our case  $b$ ) are accepted.

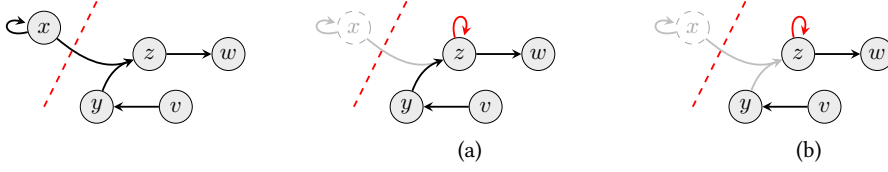


As before, we can directly exploit (ii) in order to guide the modification of  $SF_2$  to get extensions for the whole SETAF. It is however less straightforward to find a correct modification for case (iii), where a link is undecided due to what happens in  $SF_1$ .

**Example 14.** For  $SF$  (left), we have  $E = \emptyset \in \text{adm}(SF_1)$  which means  $x \in A_1 \setminus E_{R_1}^\oplus$ , i.e., the link  $(\{x, y\}, z)$  is undecided. By naively applying the same technique as in the AF case (see Example 11), we make  $z$  self-attacking. However, it is not immediately clear whether one should modify  $SF_2$  to include  $(y, z) \in R_2^*$  or not. It turns out that both options, i.e., (a) including  $(y, z)$ , and (b) not including  $(y, z)$  both lead to an undesired result.

We see that  $\{v\}$  is admissible in both case (a) and (b). However, the additional self-loop  $(z, z)$  resulting from modification makes  $z$  not acceptable in both cases. This is in contrast with the fact that  $\emptyset \cup \{v, z\}$  is indeed an admissible extension of  $SF$ .





We see that a naive generalization of the AF approach, where we blindly make those arguments that are attacked by an undecided attack self-attacking, does not work as intended. In contrast to the AF case, these attacks can still be counter-attacked if in the second part of the framework an argument of the tail is attacked—as is the case in Example 14 where  $y$  is defeated by  $v$ . In this situation,  $z$  should remain acceptable. In particular, we have to ensure that (1)  $\{z\}$  is conflict-free in the modified  $SF_2^*$ , and (2)  $z$  can only be accepted if the remaining part of the attack  $(\{x, y\}, z)$  is counter-attacked (e.g., in Example 14, if  $y$  is attacked). In fact, being *out*, the attack  $(y, z)$  is too weak and gets overwritten by the *undec* self-loop over  $z$ , thereby letting the status of  $z$  be entirely dependent on that of  $x$ . This is in opposition to what happens in the original SETAF, where the acceptance of  $z$  depends on the fact that  $y$  is defeated.

In order to present a splitting-based algorithm that works in a truly incremental and modular fashion, we consider a possible modification that is intermediate between adding an attack or not. For this, we have to make sure that the remaining part of the link is not “powerful” enough to actually defeat  $z$ —while at the same time indicating a need for  $z$  to defend against the remaining part of the attack. In the SCC-recursive schema for SETAFS [8] this issue has been resolved by marking certain attacks as *mitigated attacks*, i.e., those are attacks that have to be counter-attacked in order to accept the target argument, but cannot be used to attack the target argument in order to defend some other argument. In a nutshell, a conflict-free set is admissible if for each attack towards the set (mitigated or not) a non-mitigated counter-attack exists. However, in our modification-based approach we cannot add new syntactic concepts and adjusted semantics but have to encode this behavior within the standard SETAF syntax. A way to do this is by adding a self-attacking argument in the second sub-framework which participates in the collective attack along with the remaining part of it. This duplicate argument carries out the work of the undecided argument that is lost after splitting. Such modification can be visualized in the following example.

**Example 15.** *Contrary to the idea of mitigated attacks discussed before, we do not need labeling for attacks in this scenario. Instead, we add a dummy argument  $*$  for the undecided attack  $(\{x, y\}, z)$  such that  $*$  is self-attacking and attacks  $z$  together with  $y$  (via the attack  $(\{*, y\}, z)$ ).*

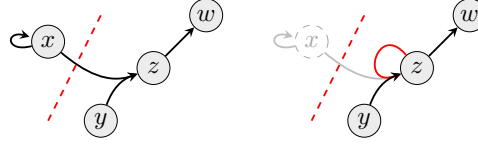


*Modifying the second part of the framework in this way successfully neutralizes the acceptance of  $w$ , which is now faced by an undecided attack. Notably, the second part of the framework is identical to the whole SETAF prior to modification.*



Such an addition can result in augmenting the number of arguments and attacks in the rightmost part of the framework. More importantly, such an unwanted outcome can be easily avoided. In fact, we can employ the very same target argument to do the job that was previously done by the dummy argument. As a consequence, the dummy argument becomes obsolete, and the modified attack collapses onto one singular (set)-self-attack on the target, as the next example illustrates.

**Example 16.** *As a final strategy we introduce a more concise and elegant modification of the second part of the framework at hand. Instead of using a dummy argument to make the attack towards  $z$  undecided, we choose to use  $z$  itself. This way, we obtain the expected result without creating a duplicate of  $x$  in the second part of the framework. This means, to account for the attack  $(\{x, y\}, z)$  with the undecided argument  $x$  in the original framework (left), we introduce in the modification the attack  $(\{y, z\}, z)$  which is a “set-self-attack” (right). Note that  $z$  is not a classic “self-attacker” as in the AF-case, since we do not introduce an attack  $(z, z)$ .*



In the modification (right), the attack  $(\{z, y\}, z)$  can never defeat its head  $z$  (since its conflicting tail  $\{y, z\}$  would have to be accepted). Since  $z$  is not defeated, it is undecided, which carries over to  $w$  via the unaltered attack  $(z, w)$ . This yields the admissible set  $\emptyset \cup \{y\}$  as desired.

As a sanity check, note that this is indeed a generalization of the AF modification. In fact, for attacks  $(T, h)$  with  $|T| = 1$ , e.g. for  $(T, h) = (\{t\}, h)$ , we have  $T \setminus \{t\} = \emptyset$ . Thus, we add the self-attack  $(\emptyset \cup \{t\}, t)$  which is the attack  $(t, t)$ .

#### 4. Reduct, SETAF Modification, and Splitting Theorem

In this section, we introduce the formal definitions that are needed to prove the correctness of our proposed splitting-based algorithm. Following Baumann [2], we generalize the notions of *reduct* and *modification*, in application to the rightmost part of the original SETAFs. Intuitively, the reduct takes care of the arguments in  $SF_2$  that are already defeated by  $E_1$  by removing them, and modifies the links that we characterize as *in*-case (1)—s.t. the remaining attack appears in the reduct  $SF'_2$ . The modification then “modifies” the undecided links by adding the targeted argument to the tail—we add (set-)self-attacks, as discussed in Example 16.

Hence, after computing an extension  $E_1$  in  $SF_1$ , we obtain the reduct of  $SF_2$  w.r.t.  $E_1$  (i.e.,  $SF'_2$ ) as follows:

1. We remove the arguments  $a \in (E_1)_{R_3}^+$  which are defeated by  $E_1$ , together with their in- and outgoing attacks, which we realize by only keeping those attacks from  $R_2$  which are completely within the new set of arguments  $A'_2$  (as in the original approach of Baumann),
2. we add the remaining part  $T \cap A'_2$  of a link  $(T, h) \in R_3$  if the tail arguments in  $T \cap A_1$  are all in  $E_1$  and no tail argument  $t \in T$  is defeated via  $R_3$  (as showcased in Example 12), given that there are any tail-arguments left in  $SF_2$  (i.e.,  $T \cap A'_2 \neq \emptyset$ ).

This allows us to retain all the information concerning defeated arguments and *in* attacks of  $SF_1$ . Formally this translates to the following notion of reduct (which we illustrate in Example 20):

**Definition 17** (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  $SF'_2 = (A'_2, R'_2)$  where,  $A'_2 = \{a \in A_2 \mid a \notin (E_1)_{R_3}^+\}$  and*

$$R'_2 = \{(T, h) \in R_2 \mid T \subseteq A'_2, h \in A'_2\} \cup \\ \{(T \cap A'_2, h) \mid (T, h) \in R_3, T \cap A'_2 \neq \emptyset, h \in A'_2, T \cap A_1 \subseteq E_1, T \cap (E_1)_{R_3}^+ = \emptyset\}$$

We have argued throughout this paper that in fact, when dealing with undecidedness, what guided our intuition towards a certain modification is not the status of the arguments in  $SF_1$ , but rather the status of the links (corresponding to cases (i)-(iii)). In fact, if we closely examine Definition 5 we can see that even in the AF case we add self-attacks to those arguments that are targeted by an undecided *link*—the set  $U_E$  of undecided arguments is merely a tool to formally obtain those links. In the context of SETAFs, where an attack is not associated to exactly one attacker, this becomes even more evident. Hence, we decide to slightly tweak the definition to omit such detour, and base our notion solely on the undecided *links*.

**Definition 18** (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:*

$$U_{R_3}^{E_1} = \{(T, h) \in R_3 \mid T \cap (E_1)_{R_1 \cup R_3}^+ = \emptyset \text{ and } \exists t \in T : t \in A_1 \setminus (E_1)_{R_1}^\oplus\}$$

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, we add to the targeted argument a (set-)self-attack incorporating the remaining part of the link (as intuitively explained in Example 16).

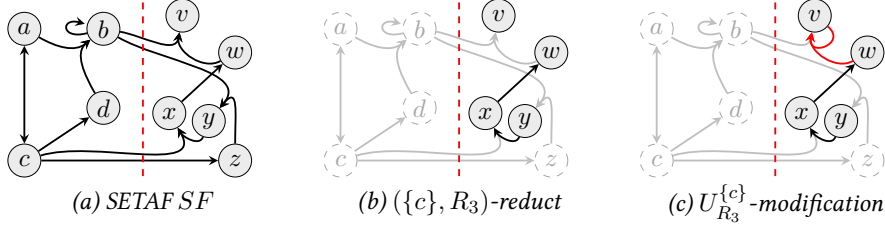
**Definition 19** (Modification). *Let  $(SF_1, SF_2, R_3)$  be a splitting for a SETAF  $SF$  and  $E_1$  an extension of  $SF_1$ . Take  $SF'_2$  as the  $(E_1, R_3)$ -reduct of  $SF_2$  and  $U_{R_3}^{E_1}$  as the set of undecided links w.r.t.  $E_1$ . We denote with  $\text{mod}_{R_3}^{E_1}(SF'_2)$  the  $U_{R_3}^{E_1}$ -modification (or simply modification) of  $SF'_2$  s.t.:*

$$\text{mod}_{R_3}^{E_1}(SF'_2) = (A'_2, R'_2 \cup \{((T \cap A'_2) \cup \{h\}, h) \mid (T, h) \in U_{R_3}^{E_1}, h \in A'_2\})$$

Before we present the main result of this paper we want to illustrate Definitions 17–19 in the following example, while covering many interesting cases at once.

**Example 20.** *In (a) we have a new SETAF  $SF$  with a splitting that separates the arguments  $A_1 = \{a, b, c, d\}$  from  $A_2 = \{v, w, x, y, z\}$ . We see that  $E_1 = \{c\}$  is admissible in the left part of the splitting. In (b) we see the reduct w.r.t. the set  $\{c\}$ , where  $a$  and  $d$  are defeated by  $c$  (as  $\{c\}_{R_1}^+ = \{a, d\}$ ) and  $b$  is undecided. This reduct contains from the right part all arguments except  $z$ , which is defeated by  $c$  (as  $\{c\}_{R_3}^+ = \{z\}$ ). We see that most attacks are removed from the right part, but  $(x, w)$  persists (since it is in  $R_2$  and all involved arguments remain), and the attack  $(\{c, y\}, x)$  is changed to  $(y, x)$ . The attack  $(\{b, z\}, y)$  is removed since  $z$  is defeated. The attack  $(\{b, w\}, v)$  is also removed, as  $b$  is undecided (i.e.,  $\{b, w\} \cap A_1 \not\subseteq E_1$ ). However, in (c) we see that*

the latter case is important for the modification: the attack  $(\{b, w\}, v)$  is an undecided link, which means in the modification we introduce the attack  $(\{v, w\}, v)$ . For the right part of the splitting we see that  $\{y, w\}$  is admissible, and obtain  $\{c, y, w\}$  as an admissible set for  $SF$ .



Note that in the first step, for the left part of the splitting, instead of the set  $\{c\}$  we could also investigate the admissible sets  $\emptyset$ ,  $\{a\}$ , or  $\{a, d\}$ , which result in different reducts and modifications.

Having these notions at hand, we now establish the adequacy of our 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 21.** *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  $SF_2^* = \text{mod}_{R_3}^{E_1}(SF_2')$ .*

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

*Proof.* (1.) We need to show for each  $(T, h) \in R_1 \cup R_2 \cup R_3$  that  $T \cup \{h\} \not\subseteq E = E_1 \cup E_2$ . Let  $SF_2' = (A_2', R_2')$  and  $SF_2^* = (A_2^*, R_2^*)$ . If  $(T, h) \in R_1$  we immediately get  $T \cup \{h\} \not\subseteq E$ , since we know  $E_1$  is conflict-free in  $SF_1$ . For  $(T, h) \in R_2$  there are two cases: either (a) the attack is removed when we construct the reduct or (b) the attack remains, i.e.,  $(T, h) \in SF_2^*$ . Case (a) happens if some  $a \in T \cup \{h\}$  is attacked by  $E_1$ , i.e.,  $(T \cup \{h\}) \cap (E_1)_{R_1 \cup R_3}^+ \neq \emptyset$ . Then at least one argument  $a \in T \cup \{h\}$  of the attack does not occur in the modification (i.e.,  $(T \cup \{h\}) \not\subseteq A_2^*$ ), and since we assume  $E_2 \in \text{cf}(SF_2^*)$  we know  $E_2 \subseteq A_2^*$ . Hence we obtain  $T \cup \{h\} \not\subseteq E$ . For case (b) we get from  $E_2 \in \text{cf}(SF_2^*)$  that at least one argument  $a \in T \cup \{h\}$  is not in  $E_2$ , which also means  $T \cup \{h\} \not\subseteq E$ . Finally, for  $(T, h) \in R_3$  we again consider two cases: (a)  $T \cap A_1 \subseteq E_1$ , and (b)  $T \cap A_1 \not\subseteq E_1$ . For case (a) we either have  $T \subseteq A_1$  in which case  $h \in (E_1)_{R_3}^+$  and we obtain  $h \notin E$  (since then  $h \notin A_2'$  while we know  $E_2 \subseteq A_2'$ ), or if  $T \not\subseteq A_1$  we get an attack  $(T \cap A_2', h) \in R_2^*$  (if otherwise  $T \cap A_2' = \emptyset$  this means we removed some  $a \in T \cap A_2$  when constructing the reduct, which means  $a \notin E_2$  and consequently  $a \notin E$ ), which since  $E_2 \in \text{cf}(SF_2^*)$  either means  $T \cap A_2' \not\subseteq E_2$  or  $h \notin E_2'$ , both give us  $T \cup \{h\} \not\subseteq E$ . For case (b) we have  $T \cap A_1 \not\subseteq E_1$ , which means  $T \cup \{h\} \not\subseteq E$ .

(2.) Suppose now that  $E \in \text{cf}(SF)$ . From this we derive that  $E \cap A_1 \in \text{cf}(SF_1)$  because every subset of a conflict-free set is also conflict-free. We now show that  $E \cap A_2 \in \text{cf}(SF_2')$ . Given that  $E \in \text{cf}(SF)$ , then for all  $T \subseteq E \cap A_1$  and  $a \in E \cap A_2$ , we have  $(T, a) \notin R_3$ . Hence, no argument in  $E$  is deleted going from  $SF_2$  to the reduct  $SF_2'$ . Thus, we conclude that  $E \cap A_2 \subseteq A_2'$ . Moreover, by  $E \in \text{cf}(SF)$  we know for each  $(T, h) \in R_2$  that  $T \cup \{h\} \not\subseteq E$

which carries over to  $SF'_2$ , since the attacks in  $R_2$  may be removed, but are never changed. Finally, whenever for a link  $(T, h) \in R_3$  with  $T \cap A_1 \subseteq E$  we add an attack  $(T \cap A'_2, h) \in R'_2$  when constructing the reduct, we also obtain  $(T \cap A'_2) \cup \{h\} \not\subseteq E$  since otherwise  $T \cup \{h\} \subseteq E$ . Therefore,  $E \cap A_2 \in cf(SF'_2)$  concluding the proof.  $\square$

Finally, we are ready to characterize the splitting algorithm by proving the main theorem of this paper for the standard Dung semantics. In particular, we show that 1. if one computes an extension  $E_1$  in  $SF_1$ , then applies the previously discussed reduct and modification, and obtains an extension  $E_2$  of the remaining sub-framework, the set-union of the two indeed make for an extension of the whole framework  $SF$ . This characterizes the incremental computation of the extension  $E$  by evaluating the two sub-frameworks. Conversely, we show that 2. if we project an arbitrary extension  $E$  of the whole framework  $SF$  to the sub-frameworks, we obtain extensions  $E_1$  for  $SF_1$  and  $E_2$  for the (w.r.t.  $E_1$ )-modified version of  $SF_2$ . This result generalizes the corresponding result of AFs [2].

Due to space constraints we present proof details only for admissible semantics, which are prototypical for the other semantics. Details for the remaining semantics can be found in [16].

**Theorem 22.** *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  $\sigma \in \{stb, adm, com, pref, grd\}$ .*

1. *If  $E_1 \in \sigma(SF_1)$  and  $E_2 \in \sigma(mod_{R_3}^{E_1}(SF'_2))$ , then  $E_1 \cup E_2 \in \sigma(SF)$ .*
2. *If  $E \in \sigma(SF)$ , then  $E \cap A_1 \in \sigma(SF_1)$  and  $E \cap A_2 \in \sigma(mod_{R_3}^{E \cap A_1}(SF'_2))$ .*

*Proof. (admissible).* (1.) Since admissibility implies conflict-freeness, we know from Proposition 21 that  $E = E_1 \cup E_2 \in cf(SF)$ . We need to show that  $E$  defends itself in  $SF$ , i.e. for all  $a \in E$ , if  $(T, a) \in R_1 \cup R_2 \cup R_3$ , then  $(T', t) \in R_1 \cup R_2 \cup R_3$  for  $T' \subseteq E$  and  $t \in T$ . Consider an argument  $a \in E_1$ .  $E_1$  defends  $a$  from each attack in  $R_1$  towards  $a$  since  $E_1 \in adm(SF_1)$ . Therefore,  $E_1 \in adm(SF)$ . Consider now an argument  $a \in E_2$  and an arbitrary attack  $(T, a) \in R_2 \cup R_3$  towards  $a$ . If  $T \cap (E_1)_{R_1 \cup R_3}^+ \neq \emptyset$  we know  $a$  is defended (in  $SF$ ) by  $E_1$  against  $(T, a)$  and we are done, hence, we proceed with the assumption  $T \cap (E_1)_{R_1 \cup R_3}^+ = \emptyset$ . This means that either  $(T \cap A'_2, a) \in R_2^*$  (via the reduct) or  $((T \cap A'_2) \cup \{a\}, a) \in R_2^*$  (via the modification). Since  $a \in E_2$  and  $E_2 \in adm(SF_2^*)$  we know there is a counter-attack in  $R_2^*$  which defends  $a$ . Even in case  $((T \cap A'_2) \cup \{a\}, a) \in R_2^*$  this counter-attack cannot be against  $a$  since this violates conflict-freeness of  $E_2$  in  $SF_2^*$ . Hence, there is some  $(S, t) \in R_2^*$  s.t.  $S \subseteq E_2$  and  $t \in T \cap A'_2$  with  $t \notin S$ . Hence, either (a)  $(S, t) \in R_2$  in which case  $a$  is defended by  $E$  in  $SF$  or (b) there is some  $(S', t) \in R_3$  with  $S' \supset S$  s.t.  $S' \cap A_1 \subseteq E_1$ , in which case  $a$  is defended (in  $SF$ ) by  $E$  via the attack  $(S', t)$  since then  $S' \subseteq E_1 \cup E_2$ . In any case we showed that  $a$  is defended in  $SF$  by  $E$ , i.e.,  $E \in adm(SF)$ .

(2.) By Proposition 21 we get  $E_1 = E \cap A_1 \in cf(SF_1)$  and  $E_2 = E \cap A_2 \in cf(SF'_2)$ . Since  $E$  defends itself in  $SF$  we get  $E \cap A_1 \in adm(SF_1)$  because  $(SF_1, SF_2, R_3)$  is a splitting of  $SF$ , i.e. no argument in  $E \cap A_1$  is attacked by a subset of  $A_2$  or defended by  $E \cap A_2$ . That is, in  $SF_1$  every attack towards an argument in  $E \cap A_1$  is countered by  $E \cap A_1$ . It remains to show that  $E \cap A_2 \in adm(SF_2^*)$ . Consider now an argument  $a \in E_2$  and an arbitrary attack  $(T, a) \in R_2^*$  against  $a$ . This attack  $(T, a)$  either corresponds to an attack  $(T, a) \in R_2$  or  $(T', a) \in R_3$  with  $T' \supset T \setminus \{a\}$  (which accounts for both the case of addition in the reduct and the modification).

In both cases we have that  $T \cap (E_1)_{R_1 \cup R_3}^+ = \emptyset$  (or  $T' \cap (E_1)_{R_1 \cup R_3}^+ = \emptyset$ , resp.) as otherwise  $(T, a)$  would not be in  $R_2^*$ . However, since  $a$  is defended by  $E$  in  $SF$ , there is a counter-attack  $(S, t) \in R_2 \cup R_3$  s.t.  $S \subseteq E$  and  $t \in (T \setminus \{a\})$  (or  $t \in (T' \setminus \{a\})$ , resp.). If  $(S, t) \in R_2$  then from  $S \subseteq E$  and  $E_2 \subseteq A_2'$  (which we get from  $E_2 \in \text{cf}(SF_2')$  via Proposition 21) and the fact that then  $(S, t) \in R_2'$  since  $S \cup \{t\} \subseteq A_2'$  we get that  $E_2$  defends  $a$  via  $(S, t)$  against  $(T, a)$  in  $SF_2^*$ . If  $(S, t) \in R_3$  since  $S \subseteq E$  we have  $S \cap A_1 \subseteq E_1$ , and hence we get an attack  $(S \cap A_1', t) \in R_2'$  which again defends  $a$  against  $(T, a)$  in  $SF_2^*$ . Hence, in every case  $a$  is defended in  $SF_2^*$ , i.e.,  $E_2 \in \text{adm}(SF_2^*)$ .  $\square$

To further establish the adequacy of our splitting approach for SETAFs, we want to highlight that we retain the close connection to the *directionality principle* [8] already proven for AFs [4]. Before introducing the definition of directionality, we first recall the notion of *influence*: in a SETAF  $SF = (A, R)$  an argument  $a \in A$  *influences* an argument  $b \in A$  if there is a path  $(a, p_1, \dots, p_n, b)$  in  $SF$  s.t. for  $1 \leq i < n$ ,  $(T_i, p_{i+1}) \in R$  with  $p_i \in T_i$ , as well as  $(T_0, p_1), (T_n, b) \in R$  with  $a \in T_0, p_n \in T_n$  holds (i.e., there is a path from  $a$  to  $b$  if we “break up” the hyperedges to standard directed edges—the *primal graph* of  $SF$ ). A set  $S \subseteq A$  is an *uninfluenced set* in  $SF$  (denoted  $S \in \text{US}(SF)$ ) if no  $a \in A \setminus S$  influences any  $b \in S$ . In other words, a set  $S$  is uninfluenced in  $SF$  if it has no incoming edges. In a nutshell, directionality states that the *projection*  $SF \downarrow_U$  of a SETAF  $SF$  to an uninfluenced set  $U$  yields the same extensions as the original framework (ignoring the arguments removed under projection).

**Definition 23.** A semantics  $\sigma$  satisfies directionality if for all SETAFs  $SF$  and every  $U \in \text{US}(SF)$  it holds  $\sigma(SF \downarrow_U) = \{E \cap U \mid E \in \sigma(SF)\}$ , where  $SF \downarrow_U = (U, \{(T', h) \mid (T, h) \in R, h \in U, T' = T \cap U, T' \neq \emptyset\})$ .

We are now able to generalize the following result regarding directionality from AFs [4, Theorem 4.13].

**Theorem 24.** Let  $\sigma$  be a semantics s.t.  $|\sigma(SF)| \geq 1$  for each SETAF  $SF$ . If  $\sigma$  allows splitting (i.e., Theorem 22 holds for  $\sigma$ ) then  $\sigma$  satisfies directionality.

*Proof.* Assume towards contradiction this is not the case, i.e., for some SETAF  $SF = (A, R)$  and some  $U \in \text{US}(SF)$  it holds  $\sigma(SF \downarrow_U) \neq \{E \cap U \mid E \in \sigma(SF)\}$ . Observe that  $(SF \downarrow_U, SF_2, R_3)$  is a splitting of  $SF$ , where  $SF_2 = (A \setminus U, R \cap (2^{A \setminus U} \times (A \setminus U)))$  and  $R_3$  contains exactly those attacks of  $SF$  that are neither in  $SF \downarrow_U$  nor  $SF_2$ .

( $\not\subseteq$ ): This means there is some  $E_1 \in \sigma(SF \downarrow_U)$  s.t.  $E_1 \neq \{E \cap U\}$  for any  $E \in \sigma(SF)$ . By Theorem 22 and since  $|\sigma(SF_2^*)| \geq 1$  we get  $E_2 \in \sigma(SF_2^*)$ , where  $SF_2^* = \text{mod}_{R_3}^{E_1}(SF_2)$  and  $SF_2'$  is the  $(E_1, R_3)$ -reduct of  $SF_2$ . Then by Theorem 22 we get  $E_1 \cup E_2 \in \sigma(SF)$ , a contradiction to the assumption that there is no  $E \in \sigma(SF)$  s.t.  $E_1 = E \cap U$ .

( $\supseteq$ ): This means there is some  $E \in \sigma(SF)$  s.t. there is no  $E_1 \in \sigma(SF \downarrow_U)$  with  $E_1 = \{E \cap U\}$ , directly contradicting Theorem 22.  $\square$

## 5. Discussion

In this paper, we introduced a modification-based splitting approach for SETAFs, and showed that it generalizes the important key features of its AF counterpart. In the following, we clarify

the relation of our splitting approach to SCC-recursiveness (as due to [8] for SETAFs) and splitting for abstract dialectal frameworks (ADFs) [3]. In the incremental computation approach induced by the SCC-recursive property, one computes the extensions in subframeworks of a given SETAF, 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. Splitting on the other hand is more general in this regard, as the subframeworks do not have to be strongly connected. Finally, SCC-recursiveness relies on a generalized semantics to deal with the decisions of prior parts of the framework, in contrast to the syntactic manipulation-based approach of splitting.

ADFs [17] 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 [18]. 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 what we discussed in Example 15, where we introduced artificial “dummy”-arguments. However, such a trick is not needed in our case, as we have illustrated.

In summary, we showed how the splitting technique can be applied in the context of collective attacks, where in contrast to the AF case also intricate situations like “diagonal splitting” can occur. We furthermore showed that the splitting theorem holds in the setting of SETAFs, and established that we retain the strong link to directionality which is known for AFs.

Our result can serve as a starting point for more general splitting ideas like parameterized splitting (cf. [5] for AFs), as well as a broader consideration in the context of dynamic argumentation. Future work includes the generalization to parameterized splitting also for SETAFs, as well as an implementation of the algorithm.

## Acknowledgments

This work has been supported by the European Union’s Horizon 2020 research and innovation programme (under grant agreement [101034440](#)). Moreover, this research has been supported by the Vienna Science and Technology Fund (WWTF) through project ICT19-065, and through the Austrian Science Fund (FWF) through projects 10.557766/COE12 and P32380.

## References

- [1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artif. Intell.* 77 (1995) 321–358.
- [2] R. Baumann, Splitting an argumentation framework, in: *Proceedings of LPNMR 2011*, volume 6645 of *LNCS*, Springer, 2011, pp. 40–53.
- [3] T. Linsbichler, Splitting abstract dialectal frameworks, in: *Proceedings of COMMA 2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2014, pp. 357–368. doi:10.3233/978-1-61499-436-7-357.



- [4] R. Baumann, *Metalogical Contributions to the Nonmonotonic Theory of Abstract Argumentation*, College Publications - Studies in Logic, 2014.
- [5] R. Baumann, G. Brewka, W. Dvořák, S. Woltran, Parameterized splitting: A simple modification-based approach, in: *Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz*, volume 7265 of *LNCS*, Springer, 2012, pp. 57–71.
- [6] S. H. Nielsen, S. Parsons, A generalization of Dung’s abstract framework for argumentation: Arguing with sets of attacking arguments, in: *Proceedings of ArgMAS 2006*, Springer, 2006, pp. 54–73. doi:10.1007/978-3-540-75526-5\_4.
- [7] G. Flouris, A. Bikakis, A comprehensive study of argumentation frameworks with sets of attacking arguments, *Int. J. Approx. Reason.* 109 (2019) 55–86. doi:10.1016/j.ijar.2019.03.006.
- [8] W. Dvořák, M. König, M. Ulbricht, S. Woltran, Principles and their computational consequences for argumentation frameworks with collective attacks, *J. Artif. Intell. Res.* 79 (2024) 69–136. doi:10.1613/JAIR.1.14879.
- [9] A. Bikakis, A. Cohen, W. Dvořák, G. Flouris, S. Parsons, Joint attacks and accrual in argumentation frameworks, in: *Handbook of Formal Argumentation*, volume 2, College Publications, 2021.
- [10] M. Caminada, M. König, A. Rapberger, M. Ulbricht, Attack semantics and collective attacks revisited, *Argument and Computation* (2024). doi:10.3233/AAC-230011, pre-press.
- [11] M. König, A. Rapberger, M. Ulbricht, Just a matter of perspective – intertranslating expressive argumentation formalisms, in: *Proceedings of COMMA 2022*, volume 353 of *FAIA*, IOS Press, 2022, pp. 212–223. doi:10.3233/FAIA220154.
- [12] Y. Dimopoulos, W. Dvořák, M. König, A. Rapberger, M. Ulbricht, S. Woltran, Redefining ABA+ semantics via abstract set-to-set attacks, in: *Proceedings of AAAI 2024*, AAAI Press, 2024, pp. 10493–10500. doi:10.1609/AAAI.V38I9.28918.
- [13] P. Baroni, G. Boella, F. Cerutti, M. Giacomin, L. W. N. van der Torre, S. Villata, On the input/output behavior of argumentation frameworks, *Artif. Intell.* 217 (2014) 144–197. doi:10.1016/j.artint.2014.08.004.
- [14] B. Liao, Toward incremental computation of argumentation semantics: A decomposition-based approach, *Ann. Math. Artif. Intell.* 67 (2013) 319–358. doi:10.1007/S10472-013-9364-8.
- [15] W. Dvořák, A. Greßler, S. Woltran, Evaluating SETAFs via answer-set programming, in: *Proceedings of SAFA 2018*, volume 2171 of *CEUR Workshop Proceedings*, 2018, pp. 10–21.
- [16] G. Buraglio, W. Dvořák, M. König, S. Woltran, Splitting Argumentation Frameworks with Collective Attacks, Technical Report DBAI-TR-2024-126, Technische Universität Wien, 2024. URL: <https://dbai.tuwien.ac.at/research/report/dbai-tr-2024-126.pdf>.
- [17] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, S. Woltran, Abstract dialectical frameworks, in: *Handbook of Formal Argumentation*, College Publications, 2018, pp. 237–285. Also appears in *IfCoLog Journal of Logics and their Applications* 4(8):2263–2318.
- [18] W. Dvořák, A. Keshavarzi Zafarghandi, S. Woltran, Expressiveness of SETAFs and support-free ADFs under 3-valued semantics, *J. Appl. Non Class. Logics* 33 (2023) 298–327. doi:10.1080/11663081.2023.2244361.