Explaining Vacuity in Abstract Argumentation

Theodoros DOUKAS, Matthias THIMM
Artificial Intelligence Group, University of Hagen, Germany

Abstract. An abstract argumentation framework is called *σ-vacuous* if it admits no non-empty extension under the semantics σ. We study how vacuity can be explained by examining how subsets of arguments and their induced substructures contribute to it. To this end, we develop a compositional account that traces how vacuity is preserved as arguments are successively added to a framework. Based on this perspective, we study the computational complexity of key tasks for constructing such explanations for admissibility-based semantics.

Keywords. Abstract argumentation, vacuity, explanations

1. Introduction

Abstract argumentation [1] provides a well-established formalism for representing and analysing conflicting information. An *argumentation framework* consists of a set of arguments equipped with a binary attack relation; an attack directed from one argument to another expresses that acceptance of the former undermines (and thus precludes) acceptance of the latter. Accordingly, an argumentation framework is naturally viewed as a directed graph, with arguments as nodes and attacks as directed edges. A *semantics* assigns to such a framework sets of arguments that can be jointly accepted in view of these attacks; these sets are called the *extensions* of the framework under the respective semantics. Different semantics have been proposed [2], and they may yield different sets of extensions for the same framework. Moreover, they may satisfy different formal properties or principles, as studied in principle-based analyses of argumentation semantics [3,4].

Recent work in abstract argumentation has investigated how the acceptability or non-acceptability of arguments can be justified and explained [5,6,7,8,9,10]. While a semantics determines extensions by definition, and thus fixes whether an argument belongs to an extension or not, explanatory approaches aim to identify structural reasons for this membership behaviour. In particular, they seek to point to syntactic features of a framework that enforce or prohibit the acceptance of specific arguments. For instance, Fan and Toni investigate explanations for both the acceptance [5] and the non-acceptance [6] of arguments in abstract argumentation frameworks; their approach characterises explanations either as admissible defence structures supporting an argument, or as minimal structures whose removal would make a previously non-acceptable argument acceptable. Moreover, Brewka *et al.* [11] analyse *strong inconsistency* for generic knowledge representation formalisms (in particular non-monotonic ones), relating *inconsistency*—i. e., the inability to derive valid information—to substructures for which inconsistency persists as they are expanded up to the entire knowledge base. Applying similar ideas to the case of abstract argumentation, Saribatur et al. [8] explain non-acceptability of arguments through *strong rejection*: an argument is strongly rejected by a substructure of the argumentation framework if the argument is rejected not only in that substructure, but also in every enclosing structure up to the original framework. The latter two approaches share a similar intuition, namely that there exist substructures that can be singularly seen as responsible for the inability to derive certain information. It is this singularity that renders both approaches insufficient as a general explanatory vehicle for non-acceptability in argumentation frameworks, as non-acceptability need not be attributable to one single cause, but rather—as we shall see—may emerge from the interaction of multiple substructures. Another explanatory approach, for the problem of explaining acceptability, is that of Bengel and Thimm [12], who provide *sequence-based explanations* for acceptance by constructing extensions step by step, thereby offering an incremental account of how acceptable sets arise. Our approach is most similar in spirit to theirs, although the details of both constructions differ significantly.

In this paper, we focus on a particular form of non-acceptability, namely σ-*vacuity* [13,14]. Given a semantics σ, a framework is σ-vacuous if it admits no non-empty extension under σ; intuitively, this happens in situations where the attack structure prevents the acceptance of any argument. Such situations may indicate modelling problems, for instance, the presence of self-attacking arguments or cyclic attack patterns. At the same time, vacuity may also arise as a genuine consequence of the interaction between arguments in highly conflicting scenarios. Understanding why a framework is vacuous is therefore important, as it can reveal structural patterns responsible for the absence of acceptable argument sets and help identify those parts of the framework that contribute to this behaviour.

Our aim is to explain σ-vacuity in structural terms by clarifying how interacting substructures give rise to vacuity of the framework as a whole. To this end, we develop a decompositional and incremental account of vacuity. Rather than seeking to identify a single structural cause, we analyse how vacuity is preserved as progressively larger parts of the framework are taken into account; these increments are the building blocks of our explanations. We introduce a measure capturing the granularity of an explanation; smaller values correspond to finer-grained explanatory steps. Finally, we show how enforcing minimality conditions on explanatory steps increases the computational complexity of constructing such explanations.

This paper makes the following contributions:

- Section 3 introduces the notion of *strong vacuity* of a subframework and investigates its usefulness as an explanatory device for vacuity of the entire framework, characterising structural conditions under which vacuity of a single component extends to the whole.

- In Section 4, we develop *vacuity-preserving decompositions* as an incremental explanation scheme, introduce a measure capturing explanatory granularity, and refine the approach by imposing minimality conditions.
- In Section 5, we analyse the complexity of associated computational tasks, locating the main decision problems for admissibility-based semantics within the polynomial hierarchy and identifying *prefix-minimality* as the source of the higher complexity.

Section 2 introduces the formal preliminaries and Section 6 concludes. Proofs of formal results can be found in an extended version of this paper.[1]

2. Preliminaries

Starting with some finite set of arguments $\mathfrak{A}$, an argumentation framework [1] is a pair $F = (\mathcal{A}, \mathcal{R})$ so that $\mathcal{A} \subseteq \mathfrak{A}$ and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. The relation $\mathcal{R}$ is the *attack relation*: given $a\mathcal{R}b$, we say that *a attacks b*. For a set of arguments $S \subseteq \mathcal{A}$ we define $S^+_F = \{a \in \mathcal{A} \mid \exists b \in S : b\mathcal{R}a\}$ and $S^-_F = \{a \in \mathcal{A} \mid \exists b \in S : a\mathcal{R}b\}$. We say that $D \subseteq \mathcal{A}$ *defends* $S \subseteq \mathcal{A}$ *(in F)* iff $S^-_F \subseteq D^+_F$. Given two sets S and S' we suggestively write $S\mathcal{R}S'$ iff $S^+_F \cap S' \neq \emptyset$; we omit braces for singleton sets, i. e., write $S\mathcal{R}a$ instead of $S\mathcal{R}\{a\}$. The *range* of a set of arguments S is defined as $S^\oplus_F = S \cup S^+_F$. Finally, $\mathfrak{F}_\mathfrak{A}$ is the set of all argumentation frameworks with arguments in $\mathfrak{A}$.

Given a framework $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_\mathfrak{A}$ and a set of arguments $S \subseteq \mathfrak{A}$, we denote with $F|_S$ the *restriction of F to S*, where $F|_S = (\mathcal{A}_S, \mathcal{R}_S)$, $\mathcal{A}_S = \mathcal{A} \cap S$ and $\mathcal{R}_S = (S \times S) \cap \mathcal{R}$. A framework $F' = (\mathcal{A}', \mathcal{R}')$ is a *subframework of F* if $\mathcal{A}' \subseteq \mathcal{A}$ and $F|_{\mathcal{A}'} = F'$; we write $F' \subseteq F$. Conversely, if $F' \subseteq F$ we also say that F is a *superframework of F'*, and write $F \supseteq F'$.

A *semantics* σ is a mapping $\mathfrak{F}_\mathfrak{A} \to 2^{2^\mathfrak{A}}$ so that $\sigma(F) \subseteq 2^\mathcal{A}$ for each $F \in \mathfrak{F}_\mathfrak{A}$, assigning to a framework those argument sets that jointly represent an acceptable point of view in the scenario described by F [1,3]. These jointly acceptable sets of arguments are called the *σ-extensions* of F.

Definition 1. For a framework $F = (\mathcal{A}, \mathcal{R})$, we define the semantics:

- $\mathsf{cf}(F) = \{S \subseteq \mathcal{A} \mid S^+_F \subseteq \mathcal{A} \setminus S\}$ (*conflict-free*);
- $\mathsf{adm}(F) = \{S \in \mathsf{cf}(F) \mid S^-_F \subseteq S^+_F\}$ (*admissible*);
- $\mathsf{com}(F) = \{S \in \mathsf{adm}(F) \mid \forall a \in \mathcal{A} : \{a\}^-_F \subseteq S^+_F \implies a \in S\}$ (*complete*);
- $\mathsf{pr}(F) = \{S \in \mathsf{adm}(F) \mid \nexists S' \in \mathsf{adm}(F) : S \subsetneq S'\}$ (*preferred*);
- $\mathsf{naive}(F) = \{S \in \mathsf{cf}(F) \mid \nexists S' \in \mathsf{cf}(F) : S \subsetneq S'\}$ (*naive*);
- $\mathsf{sstb}(F) = \{S \in \mathsf{com}(F) \mid \nexists S' \in \mathsf{com}(F) : S^\oplus_F \subsetneq S'^\oplus_F\}$ (*semi-stable*).

The following notion of *defence* is fundamental to admissible sets, which are exactly those conflict-free argument sets that defend all their elements.

---

[1] `https://zenodo.org/records/19563597`

Definition 2. A semantics σ is *conflict-free* if $\sigma(F) \subseteq \mathsf{cf}(F)$ for all $F \in \mathfrak{F}_{\mathfrak{A}}$, and it satisfies *defence* if for all $F \in \mathfrak{F}_{\mathfrak{A}}$ and $S \in \sigma(F)$ we have $S^-_F \subseteq S^+_F$. It satisfies *admissibility* if it satisfies both conflict-freeness and defence.

All semantics from Definition 1 are conflict-free and all except cf and naive are admissible.

We say that a framework F is *σ-inconsistent* (for some semantics σ) if $\sigma(F) = \emptyset$. A framework F is *σ-vacuous* [13,14] if its σ-extensions contain at most the empty set, i. e., $\sigma(F) \subseteq \{\emptyset\}$.

Example 1. The framework $F = (\{a\}, \{(a, a)\})$ is adm-vacuous, but not adm-inconsistent, since $\mathsf{adm}(F) = \{\emptyset\}$.

3. Strong Vacuity

In this section, we investigate a strengthened notion of vacuity as a means of tracing the vacuity of a framework to a specific subcomponent. The guiding idea is to capture situations in which a part of the framework enforces vacuity in a way such that the absence of non-empty extensions can be attributed singularly to that component.

Subsection 3.1 introduces the notion formally and situates it within a broader conceptual context. In Subsection 3.2, we analyse a structurally simple case in which the vacuity of a single subframework can be identified as the cause of global vacuity: namely, when an unattacked subframework itself attacks all arguments outside it.

3.1. Strong Inconsistency and Strong Vacuity

We now introduce the notion of *strong inconsistency*; for that, following Brewka *et al.* [11], we define a *logic* in an abstract manner.

Definition 3 (Logic [11])**.** A *logic* $\mathcal{L} = (\Phi, \mathsf{B}, \mathsf{I}, f_{\mathrm{acc}})$ is a quadruple consisting of a set of *well-formed formulas* Φ, a set of *belief sets* B, an upward-closed[2] set of *inconsistent belief sets* $\mathsf{I} \subseteq \mathsf{B}$, and an *acceptability function* $f_{\mathrm{acc}} : 2^{\Phi} \to 2^{\mathsf{B}}$. A *knowledge base K of $\mathcal{L}$* is a finite subset of Φ; K is *inconsistent* iff $f_{\mathrm{acc}}(K) \subseteq \mathsf{I}$.

A strong notion of inconsistency is then defined as follows.

Definition 4 (Strong Inconsistency [11])**.** Let $K_0 \subseteq \Phi$. A knowledge base $K \subseteq K_0$ is *strongly K_0-inconsistent* if for every K' with $K \subseteq K' \subseteq K_0$ we have that $f_{\mathrm{acc}}(K') \subseteq \mathsf{I}$.

We adapt this notion to the knowledge representation formalism of abstract argumentation and specialise it to vacuity as follows.[3] Fix a framework $F_0 = (\mathcal{A}_0, \mathcal{R}_0)$ and a semantics σ, and set $\Phi = \mathcal{A}_0$, $\mathsf{B} = 2^{\mathcal{A}_0}$, and $\mathsf{I} = \emptyset$. For

---

[2] A set of sets $\mathbb{S}$ is *upward-closed* iff $S \in \mathbb{S}$ and $S \subseteq S'$ implies $S' \in \mathbb{S}$.

[3] Brewka *et al.* [11] give a related instantiation for inconsistency and stable semantics.

every knowledge base $K \subseteq \mathcal{A}_0$, set $f_{\text{acc}}(K) = \sigma(F_0|_K) \setminus \{\emptyset\}$. Then K represents an argumentation framework via the mapping $K \mapsto F_0|_K$. Since $f_{\text{acc}}(K) \subseteq \mathsf{I}$ iff $\sigma(F_0|_K) \subseteq \{\emptyset\}$, inconsistent knowledge bases in the terminology of the abstract logic correspond exactly to σ-vacuous argumentation frameworks. Strong inconsistency therefore yields the following notion of *strong vacuity*.

Definition 5 (Strong Vacuity)**.** Let $F_0 = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$ and let σ be a semantics. The framework $F \subseteq F_0$ is *strongly σ-vacuous w.r.t. F_0* if every superframework $F' \supseteq F$ with $F' \subseteq F_0$ is σ-vacuous.

Example 2. Consider the framework (F) in Figure 1, where $F|_{\{a\}}$ consists of the self-attacking argument a and thus is adm-vacuous; it is not adm-inconsistent, since the empty set remains admissible in every framework. Expanding $F|_{\{a\}}$ by incorporating arguments b and c, we find that in fact every superframework of $F|_{\{a\}}$, up to F, is adm-vacuous; thus, $F|_{\{a\}}$ is strongly adm-vacuous.

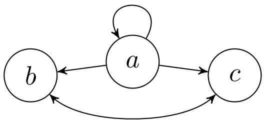

Figure 1. This framework contains a minimal strongly vacuous subframework.

Saribatur *et al.* [8] introduce the related notion of *strong rejection*, which concerns the persistent non-acceptability of a single argument under expansion.

Definition 6. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$, let σ be a semantics, let $S \subseteq \mathcal{A}$, and let $a \in \mathcal{A}$. The subframework $F|_S$ *strongly rejects a w.r.t. F under σ* if for each $S' \subseteq \mathcal{A}$ with $S \subseteq S'$ we have $a \notin \bigcup \sigma(F|_{S'})$.

Example 3. In the framework (F) of Figure 1, the subframework $F|_{\{a\}}$ strongly rejects all of a, b, c under adm.

Strong vacuity is in fact equivalent to strong rejection of every argument.

Proposition 1. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$, let σ be a semantics, and let $S \subseteq \mathcal{A}$. The subframework $F|_S$ is strongly σ-vacuous w.r.t. F iff $F|_S$ strongly rejects every $a \in \mathcal{A}$ w.r.t. F under σ.

3.2. Vacuity in Unattacked Subframeworks

Example 2 illustrates how vacuity of a subframework may carry over to the full framework: a subframework that is vacuous in isolation may force every enclosing framework to remain vacuous as well. We now study this for unattacked vacuous subframeworks, assuming that every framework with a non-empty admissible set also has a non-empty σ-extension. The following notion captures this condition.

Definition 7. A semantics σ is adm-*responsive* if, for any framework $F \in \mathfrak{F}_{\mathfrak{A}}$, $\sigma(F) \nsubseteq \{\emptyset\}$ whenever $\mathsf{adm}(F) \nsubseteq \{\emptyset\}$.

Lemma 1. adm, pr, com, sstb, and naive are adm-responsive.

In the following, let $\mathbb{U}(F) = \{U \subseteq \mathcal{A} \mid (\mathcal{A} \setminus U)^+_F \cap U = \emptyset\}$ be the *unattacked* argument sets, i. e., sets of arguments that are not attacked from outside.

Proposition 2. Let σ be an adm-responsive semantics that satisfies admissibility, and let F be a σ-vacuous framework. Then for any $U \in \mathbb{U}(F)$, $F|_U$ is σ-vacuous as well.

We then ask when vacuity of an unattacked subframework implies vacuity of the entire framework.

Proposition 3. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$, let σ be a semantics that satisfies adm-responsiveness and admissibility, and let $U \in \mathbb{U}(F)$ such that $(\mathcal{A} \setminus U) \subseteq U^+_F$. If $F|_U$ is σ-vacuous, then so is F.

The above result shows that, in the particular case of an unattacked vacuous subframework, it is sufficient for the subframework to attack all of its surrounding arguments for vacuity to carry over to the entire framework.

Example 4. In the framework (F) of Figure 1, we have $\{a\} \in \mathbb{U}(F)$, and a attacks both b and c. Consequently, F is $\{\mathsf{adm}, \mathsf{pr}, \mathsf{com}, \mathsf{sstb}\}$-vacuous. Since naive does not satisfy admissibility, we cannot infer vacuity of F for that semantics; and indeed, both $\{b\}$ and $\{c\}$ are naive extensions of F.

The unattacked and vacuous subframeworks of Proposition 3 that attack all of their surroundings are actually strongly vacuous. Indeed, attacking all (not self-attacking) surrounding arguments is necessary for strong vacuity.[4]

Proposition 4. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$, let σ be an adm-responsive semantics, and let $S \subseteq \mathcal{A}$. If $F|_S$ is strongly σ-vacuous, then either $a\mathcal{R}a$ or $S\mathcal{R}a$ for all $a \in \mathcal{A} \setminus S$.

Propositions 3 and 4 characterise strong vacuity of unattacked subframeworks as follows.

Corollary 1. Let $F \in \mathfrak{F}_{\mathfrak{A}}$, let σ be an adm-responsive semantics that satisfies admissibility, and let $S \in \mathbb{U}(F)$. The subframework $F|_S$ is strongly σ-vacuous iff it is σ-vacuous, and $S\mathcal{R}a$ or $a\mathcal{R}a$ for all $a \in \mathcal{A} \setminus S$.

For unattacked subframeworks, strong vacuity imposes a highly restrictive structural condition: the vacuous component must attack every non-self-attacking argument outside it. This limitation is of course the immediate consequence of wanting to attribute global vacuity to one singular cause, and it motivates the more general decompositional account developed next.

---

[4] Saribatur *et al.* state a similar proposition w.r.t. strong rejection of single arguments under admissible and grounded semantics [8].

4. Vacuity-Preserving Decompositions

The approach taken in this section is inspired by *serialisability* [15] for admissibility-based semantics, where extensions are constructed stepwise, by selecting *initial sets* (i. e., $\subseteq$-minimal, non-empty admissible sets [16]), removing them together with the arguments they attack, and continuing on the remaining framework. In a similar spirit, we analyse vacuity incrementally: starting from a $\subseteq$-minimal vacuous subframework, we successively add arguments and their induced attacks while requiring vacuity to be preserved at every stage. This yields *vacuity-preserving decompositions* as a formal account of stepwise explanations of vacuity (Section 4.1); Section 4.2 then refines this account by imposing minimality conditions on the increments.

4.1. Definition and Basic Properties

We now show how a vacuous framework can be decomposed so as to yield a stepwise explanation of vacuity.

Definition 8 (Decomposition of a framework)**.** Let σ be a semantics and let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$ be σ-vacuous. A *σ-vacuity-preserving decomposition of F* is a finite sequence $\langle G_1, \ldots, G_n \rangle$ of pairwise disjoint, non-empty subsets of $\mathcal{A}$, such that $\bigcup_{i=1}^{n} G_i = \mathcal{A}$, and for $1 \leq i \leq n$, $\sigma(F|_{S_i}) \subseteq \{\emptyset\}$ where $S_i = \bigcup_{j=1}^{i} G_j$.

A σ-vacuity-preserving decomposition of a framework partitions its argument set into components, such that σ-vacuity is preserved as the framework is stepwise assembled along the decomposition.

Example 5. The framework in Figure 2 admits the following adm-vacuity-preserving decompositions: $D_1 = \langle \{d\}, \{a, b, c\}, \{e\} \rangle$, $D_2 = \langle \{d\}, \{a, b, c, e\} \rangle$, $D_3 = \langle \{a, b, c\}, \{d\}, \{e\} \rangle$, $D_4 = \langle \{a, b, c\}, \{e\}, \{d\} \rangle$, $D_5 = \langle \{a, b, c\}, \{d, e\} \rangle$, $D_6 = \langle \{a, b, c, d, e\} \rangle$.

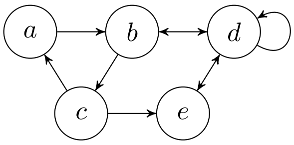

Figure 2. This framework admits several vacuity-preserving decompositions.

Intuitively, smaller components provide finer explanations, motivating the following measure.

Definition 9. For a sequence of sets $D = \langle G_1, \ldots, G_n \rangle$, we define its *width*

$$|D|_{\max} = \max_{1 \leq i \leq n} |G_i|.$$

For the empty sequence, we set $|\langle\rangle|_{\max} = 0$.

Applied to a σ-vacuity-preserving decomposition of F, its width measures the size of the largest component that must be added at once in order to preserve vacuity. We will refer to this measure as the *σ-vacuity width* of the decomposition.

Example 6. Using the designations from Example 5, we have $|D_1|_{\max} = 3$, $|D_2|_{\max} = 4$, $|D_3|_{\max} = |D_4|_{\max} = |D_5|_{\max} = 3$, and $|D_6|_{\max} = 5$.

In what follows, we often consider decompositions not of the whole argument set, but relative to a fixed initial component. This reflects the incremental nature of our approach: at each stage, vacuity is evaluated with respect to the arguments already introduced.

Definition 10 (Decomposition with respect to a prefix)**.** Let σ be a semantics, let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$ be σ-vacuous, and let $S \subseteq \mathcal{A}$ such that $F|_S$ is σ-vacuous. A *σ-vacuity-preserving decomposition of $\mathcal{A} \setminus S$* is a finite (and possibly empty) sequence $\langle G_1, \ldots, G_n \rangle$ of pairwise disjoint, non-empty subsets of $\mathcal{A} \setminus S$, such that $\langle S, G_1, \ldots, G_n \rangle$ is a σ-vacuity-preserving decomposition of F when $S \neq \emptyset$, and $\langle G_1, \ldots, G_n \rangle$ is such a decomposition when $S = \emptyset$.

Example 7. Going back to the framework $F = (\mathcal{A}, \mathcal{R})$ in Figure 2, we find that the non-empty and $\subseteq$-minimal adm-vacuous subframeworks of F are induced by $S_1 = \{d\}$ and $S_2 = \{a, b, c\}$. Both $D'_1 = \langle \{a, b, c\}, \{e\} \rangle$ and $D'_2 = \langle \{a, b, c, e\} \rangle$ are adm-vacuity-preserving decompositions of $\mathcal{A} \setminus S_1$, and $D'_3 = \langle \{d\}, \{e\} \rangle$, $D'_4 = \langle \{e\}, \{d\} \rangle$, $D'_5 = \langle \{d, e\} \rangle$ are adm-vacuity-preserving decompositions of $\mathcal{A} \setminus S_2$.

Our construction relates to strong vacuity (Definition 5) as follows.

Corollary 2. Let $F = (\mathcal{A}, \mathcal{R})$ be a framework, σ a semantics, and let $S \subseteq \mathcal{A}$ such that $F|_S$ is σ-vacuous. Then $F|_S$ is strongly σ-vacuous if and only if every ordered partition of $\mathcal{A} \setminus S$ into pairwise disjoint, non-empty subsets is a σ-vacuity-preserving decomposition of $\mathcal{A} \setminus S$.

Given a vacuous prefix S, we measure the width of decompositions extending S to the full framework.

Definition 11 (Vacuity width relative to a prefix)**.** Let σ be a semantics so that $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$ is σ-vacuous, and let $S \subseteq \mathcal{A}$. Provided $F|_S$ is σ-vacuous, the *σ-vacuity width of F relative to S* is the minimum width over all σ-vacuity-preserving decompositions of $\mathcal{A} \setminus S$.

Example 8. Continuing Example 7, we find $|D'_1|_{\max} = 3$ and $|D'_2|_{\max} = 4$, so that the adm-vacuity width of F relative to S_1 is 3. Moreover, we have that $|D'_3|_{\max} = |D'_4|_{\max} = 1$ and $|D'_5|_{\max} = 2$, therefore the adm-vacuity width of F relative to S_2 is 1.

Corollary 3. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$, let $S \subseteq \mathcal{A}$, and let σ be a semantics. If $F|_S$ is strongly σ-vacuous, then the σ-vacuity width of F relative to S is at most 1.

The next proposition establishes a monotonicity property of vacuity width relative to a prefix under restriction to unattacked parts: restricting to such parts cannot increase the width, so they provide local lower bounds on the granularity of explanations in the full framework.

Proposition 5. Let σ be an adm-responsive semantics that satisfies admissibility. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$ be σ-vacuous, let $S \subseteq \mathcal{A}$ so that $F|_S$ is σ-vacuous, and let $U \in \mathbb{U}(F)$. Then the σ-vacuity width of F relative to S is greater than or equal to the σ-vacuity width of $F|_U$ relative to $S \cap U$.

The final example in this subsection shows how a vacuity-preserving decomposition can be used to explain vacuity, proceeding stepwise along the members of the decomposition sequence.

Example 9. Let us explain the adm-vacuity of the framework (F) in Figure 2, starting with $S = \{d\}$ and a decomposition $D = \langle \{a, b, c\}, \{e\} \rangle$. First, $F|_S$ is vacuous since d attacks itself. Second, the odd cycle $\{a, b, c\}$ is vacuous, and the self-attacking d cannot render any of its subsets admissible. Third, e's attack on d is insufficient to support an admissible set containing e, so vacuity persists.

4.2. Minimality

For explanatory purposes, it is natural to focus on substructures that are vacuous in isolation and on increments that are minimal relative to a given prefix. We formalise these notions below.

Definition 12 (Minimal vacuity)**.** Given a semantics σ, a non-empty framework $F = (\mathcal{A}, \mathcal{R})$ ($\mathcal{A} \neq \emptyset$) is *minimally σ-vacuous* if it is σ-vacuous, and no non-empty, strict subframework $F|_S$ with $\emptyset \neq S \subsetneq A$ is σ-vacuous.

Example 10. In the framework (F) of Figure 2, $F|_{\{a,b,c\}}$ as well as $F|_{\{d\}}$ are minimally adm-vacuous.

The second notion concerns minimality of increments relative to a prefix: to identify only the relevant elements of an explanation, each increment should be minimal relative to the current prefix.

Definition 13. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$, let σ be a semantics, let $S \subseteq \mathcal{A}$ such that $F|_S$ is σ-vacuous, and let $G \subseteq \mathcal{A} \setminus S$. We say that $F|_G$ is *prefix-minimally σ-vacuous with respect to F and S* if $F|_{S \cup G}$ is σ-vacuous, and $F|_{S \cup G'}$ is not σ-vacuous for any $\emptyset \subsetneq G' \subsetneq G$.

Example 11. In the framework (F) from Figure 2, $F|_{\{d\}}$ is prefix-minimally adm-vacuous with respect to F and $\{a, b, c\}$. At the same time, $F|_{\{a,b,c\}}$ is prefix-minimally adm-vacuous with respect to F and $\{d\}$.

Definitions 11 and 12 together give rise to the following notion.

Definition 14 (Vacuity width of a framework)**.** For a semantics σ and a σ-vacuous framework $F = (\mathcal{A}, \mathcal{R})$, the *$\sigma$-vacuity width of* F is the minimal σ-vacuity width of F with respect to S, taken over all minimally σ-vacuous prefixes $S \subseteq \mathcal{A}$.

Thus if the σ-vacuity width of F is k, then every explanation of its vacuity that starts from a minimally σ-vacuous prefix must contain a step in which at least k arguments are introduced simultaneously.

The minimal vacuous subframeworks provide natural starting points for an explanation of vacuity. From there, we seek decompositions that are not only width-minimal—since width reflects only the largest increment—but whose steps are prefix-minimally vacuous with respect to their prefixes. Requiring this for every step in the decomposition yields the following notion.

Definition 15. Let σ be a semantics, let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$ be σ-vacuous, and let $S \subseteq \mathcal{A}$ such that $F|_S$ is σ-vacuous. A σ-vacuity-preserving decomposition $\langle G_1, \ldots, G_n \rangle$ of $\mathcal{A} \setminus S$ is *prefix-minimal* if for each G_i, $F|_{G_i}$ is prefix-minimally σ-vacuous with respect to F and $S \cup G_1 \cup \ldots \cup G_{i-1}$.

A vacuity-preserving decomposition can always be rendered prefix-minimal.

Proposition 6. Let $F = (\mathcal{A}, \mathcal{R}) \in \mathfrak{F}_{\mathfrak{A}}$, let σ be a semantics, and let $S \subseteq \mathcal{A}$ such that $F|_S$ is σ-vacuous. Every non-empty $G \subseteq \mathcal{A} \setminus S$ for which $F|_{S \cup G}$ is σ-vacuous contains a non-empty subset $G' \subseteq G$ so that $F|_{G'}$ is prefix-minimally σ-vacuous with respect to F and S.

Vacuity-preserving decompositions explain vacuity by tracing how it persists along successive steps. Width captures the size of the largest increment, i.e., the most complex explanatory step, while prefix-minimality isolates those increments that are essential at each stage. Our decompositions also admit a temporal interpretation: arguments are revealed stepwise, and vacuity must persist throughout.

5. Computational Complexity

We now turn to the computational complexity of constructing explanations. We assume familiarity with basic complexity theory and the polynomial hierarchy; for an introduction, see [17]. We begin by recalling the following result due to Dvořák and Dunne [18] for the semantics relevant to our setting.

Proposition 7. Determining whether a framework F admits a nonempty σ-extension is NP-complete for $\sigma \in \{\mathsf{adm}, \mathsf{pr}, \mathsf{com}, \mathsf{sstb}\}$.

The following decision problems are relevant to our approach.

Problem 1 (VPD-VERIFY)**.** Given a framework $F = (\mathcal{A}, \mathcal{R})$ and $S \subseteq \mathcal{A}$, decide whether a decomposition $\langle G_1, \ldots, G_n \rangle$ of $\mathcal{A} \setminus S$ is σ-vacuity-preserving.

Problem 2 (PREFIX-MINIMAL-VERIFY)**.** Given a framework $F = (\mathcal{A}, \mathcal{R})$, a prefix $S \subseteq \mathcal{A}$, and an increment $G \subseteq \mathcal{A} \setminus S$, decide whether $F|_G$ is prefix-minimally σ-vacuous with respect to F and S.

Problem 3 (WIDTH)**.** Given a framework F and an integer $k \geq 0$, decide whether F is σ-vacuous and the σ-vacuity width of F is at most k.

Problem 4 (STRONG-VACUOUS-VERIFY)**.** Given frameworks F_0 and $F \subseteq F_0$, decide whether F is strongly σ-vacuous with respect to F_0.

Problem 5 (Min-Vacuous-Verify)**.** Decide whether a framework is minimally σ-vacuous.

Proposition 8. The complexity of the above problems for the case $\sigma = \mathsf{adm}$ is located within the polynomial hierarchy as follows.

1. Vpd-Verify is coNP-complete.
2. Prefix-Minimal-Verify is coNP-hard and belongs to Π_2^P.
3. Width is coNP-hard and belongs to Σ_3^P.
4. Strong-Vacuous-Verify is coNP-complete.
5. Min-Vacuous-Verify belongs to Π_2^P.

For Prefix-Minimal-Verify we obtain coNP-hardness and membership in Π_2^P; for Min-Vacuous-Verify we currently only obtain membership in Π_2^P. Whether either problem is Π_2^P-complete remains open.[5] Observe that Min-Vacuous-Verify is a special case of Prefix-Minimal-Verify (take $S = \emptyset$ and $G = \mathcal{A}$), so any hardness result for Min-Vacuous-Verify will immediately transfer to Prefix-Minimal-Verify.

Regarding the construction of explanations, the computational hardness does not arise from checking vacuity itself—decidable in coNP—but from enforcing prefix-minimality. The latter requires quantification over all strict subincrements, thereby lifting the problem to the second level of the polynomial hierarchy.

A note on generality. The above membership arguments depend only on the complexity of deciding whether a framework admits a non-empty σ-extension. If this underlying decision problem belongs to some complexity class C, then the same reasoning yields corresponding upper bounds expressed relative to an oracle access to C: Vpd-Verify and Strong-Vacuous-Verify lie in $\mathsf{coNP}^{\mathsf{C}}$, Prefix-Minimal-Verify and Min-Vacuous-Verify lie in $\Pi_2^{P,\mathsf{C}}$, and Width lies in $\Sigma_3^{P,\mathsf{C}}$. Here $\Pi_2^{P,\mathsf{C}}$ and $\Sigma_3^{P,\mathsf{C}}$ denote the second resp. third level of the polynomial hierarchy with oracle access to C, meaning that machines at this level may query the non-emptiness decision problem as an oracle in addition to their usual polynomial-time and NP-type reasoning.

For the semantics of interest to us, Proposition 7 establishes that non-emptiness of σ-extensions has the same complexity as for adm; together with the constructions above, this yields the identical bounds.

Proposition 9. For $\sigma \in \{\mathsf{adm}, \mathsf{pr}, \mathsf{com}, \mathsf{sstb}\}$, we have that both Vpd-Verify and Strong-Vacuous-Verify lie in coNP, Prefix-Minimal-Verify and Min-Vacuous-Verify lie in Π_2^P, and Width lies in Σ_3^P.

6. Discussion

In this paper, we introduced vacuity-preserving decompositions as a compositional and incremental framework for explaining σ-vacuity in abstract argumentation.

---

[5] In particular, showing that either problem is even NP-hard has proven elusive, since they deal with *non*-existence of (non-empty) extensions; however, Π_2^P-membership is also the best upper bound we could find.

Rather than attributing vacuity to a single substructure, our approach captures how vacuity emerges from the interaction of multiple components by tracing its preservation along a sequence of argument additions. We defined vacuity width as a measure that quantifies the granularity of such explanations and analysed its structural properties, including its relation to strong vacuity and its behaviour under restriction to unattacked subframeworks. We further refined the framework by introducing minimality conditions on explanatory steps, ensuring that each increment captures an essential contribution to vacuity. Finally, we investigated the computational complexity of constructing such explanations for admissibility-based semantics, showing that while basic verification tasks lie at the first level of the polynomial hierarchy, enforcing minimality raises the complexity to the second and third level.

Acknowledgments. The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (project “Argumentative Reasoning in Nonsensical Situations”, grant 550735820).

References

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

[2] Baroni P, Caminada M, Giacomin M. An introduction to argumentation semantics. The knowledge engineering review. 2011;26(4):365-410.

[3] Baroni P, Giacomin M. On principle-based evaluation of extension-based argumentation semantics. Artif Intell. 2007;171(10-15):675-700.

[4] van der Torre L, Vesic S. The principle-based approach to abstract argumentation semantics. IfCoLog Journal of Logics and Their Applications. 2017;4:2735-78.

[5] Fan X, Toni F. On computing explanations in argumentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29; 2015. p. 1496-502.

[6] Fan X, Toni F. On explanations for non-acceptable arguments. In: International Workshop on Theory and Applications of Formal Argumentation. Springer; 2015. p. 112-27.

[7] Liao B, van der Torre L. Explanation semantics for abstract argumentation. In: Computational Models of Argument: Proceedings of COMMA 2020; 2020. p. 271-82.

[8] Saribatur ZG, Wallner JP, Woltran S. Explaining non-acceptability in abstract argumentation. In: ECAI 2020. IOS Press; 2020. p. 881-8.

[9] Borg A, Bex F. A basic framework for explanations in argumentation. IEEE Intelligent Systems. 2021;36(2):25-35.

[10] Ulbricht M, Wallner JP. Strong explanations in abstract argumentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35; 2021. p. 6496-504.

[11] Brewka G, Thimm M, Ulbricht M. Strong inconsistency. Artif Intell. 2019;267:78-117.

[12] Bengel L, Thimm M. Sequence explanations for acceptance in abstract argumentation. In: Proceedings of KR 2025; 2025. p. 121–131.

[13] Thimm M. On undisputed sets in abstract argumentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37; 2023. p. 6550-7.

[14] Blümel L, Thimm M. Revisiting Vacuous Reduct Semantics for Abstract Argumentation. In: Proc. of the 27th European Conf. on Artificial Intelligence (ECAI’24); 2024. p. 3517-24.

[15] Thimm M. Revisiting initial sets in abstract argumentation. Argument & Computation. 2022;13(3):325-60.

[16] Xu Y, Cayrol C. Initial sets in abstract argumentation frameworks. Journal of Applied Non-Classical Logics. 2018;28(2-3):260-79.

[17] Papadimitriou CH. Computational complexity. In: Encyclopedia of computer science; 2003. p. 260-5.

[18] Dvorák W, Dunne PE. Computational problems in formal argumentation and their complexity. Journal of Applied Logics. 2017;4(8):2557-622.