Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Abstraction in Assumption-based Argumentation

Iosif Apostolakis[1] , **Zeynep G. Saribatur**[2] , **Johannes P. Wallner**[1]
[1]Institute of Software Technology, TU Graz
[2]Institute of Logic and Computation, TU Wien
apostolakis@ist.tugraz.at, zeynep@kr.tuwien.ac.at, wallner@ist.tugraz.at

Abstract

Approaches to computational argumentation provide foundational ways to reason argumentatively within Artificial Intelligence (AI). The underlying formal approaches can oftentimes be classified into structured argumentation and abstract argumentation. The former prescribe rigorous workflows, starting from knowledge bases to finding arguments in favour and against claims under scrutiny, and drawing conclusions. Abstract argumentation provides formal semantics operating on arguments whose internal structure is hidden and only relations are kept for reasoning, resulting in so-called argumentation frameworks (AFs). In this work, we apply a form of existential abstraction on the prominent structured approach of assumption-based argumentation (ABA), leading to an interactive way of simplifying argumentation scenarios by abstracting irrelevant details, towards supporting explainability. Existential abstraction was shown to be promising in many areas of AI, including a recent work on AFs. We lift this approach to the structured level—which is, as we show, both not direct from AFs and can benefit from utilization of the internal structure of arguments. Among our contributions, we introduce existential abstraction on ABA via clustering assumptions, develop semantics on clustered ABA frameworks for reasoning on such clusterings, show differences to the level of AFs, and provide a prototype interactive tool that obtains faithful clusterings that do not lead to any spurious reasoning.

1 Introduction

Computational argumentation is a well-established area within Artificial Intelligence (AI) that provides foundational approaches to dialectical and argumentative reasoning (Baroni et al. 2018; Gabbay et al. 2021), with a variety of heterogeneous application areas (Atkinson et al. 2017) such as legal reasoning (Prakken and Sartor 2015), medical reasoning (Cyras et al. 2021a), and multi-agent systems (Amgoud, Dimopoulos, and Moraitis 2007; Dimopoulos, Mailly, and Moraitis 2019; Fan and Toni 2012).

Central to approaches in this field are formalisations of argumentation, which can be categorized into structured argumentation (Čyras et al. 2018; Modgil and Prakken 2018; García and Simari 2018; Besnard and Hunter 2018; Gordon, Prakken, and Walton 2007; Kakas, Moraitis, and Spanoudakis 2019) and abstract argumentation (Dung 1995; Baroni et al. 2018). The former provides principled approaches of how to reason argumentatively on given, possibly conflicting, knowledge bases. This is achieved by prescribing how to instantiate argument structures and their relationships from given knowledge. Abstract argumentation, on the other hand, provides approaches on how to find acceptable (sets of) arguments when arguments are seen as entities without internal structure. The most prominent formalisation in abstract argumentation being argumentation frameworks (AFs) (Dung 1995), which represent arguments as vertices and a directed counter-argument (attack) relation between arguments is represented as directed edges. Acceptance of sets of arguments is defined through argumentation semantics (Baroni, Caminada, and Giacomin 2011).

Inherently, one central aim of computational argumentation is to provide argumentative reasons, explanations, or explications for statements that are under scrutiny. Naturally, over the course of the last decades several approaches were studied that augment argumentative methods for explainability or supporting explanations, see also two recent surveys for an overview on this topic (Cyras et al. 2021b; Vassiliades, Bassiliades, and Patkos 2021).

Among the methods supporting explainability, we in particular find approaches that simplify given argumentation scenarios (Baroni et al. 2014; Fan and Toni 2015; Ulbricht and Wallner 2022; Sakama 2018; Dvořák et al. 2019), with a recent approach utilizing abstraction on AFs via clustering of arguments (Saribatur and Wallner 2021). As witnessed by several works in the field, a lifting or adaptation of approaches from abstract to structured argumentation is both not direct (Prakken and Winter 2018; Wallner 2020; Prakken 2023; Rapberger and Ulbricht 2023) and brings such approaches closer to applications.

In this paper we introduce existential abstraction to structured argumentation by clustering (parts of) given knowledge bases and thus lift abstraction to the level of structured argumentation.

Abstraction can be useful to focus on relevant details, abstracting away redundant parts or undesired parts. Our approach also provides an automated way with possible user-interaction of applying existential abstraction. We focus on the prominent structured approach of (flat) assumption-based argumentation (ABA) (Čyras et al. 2018), with applications in, e.g., medical reasoning (Cyras et al. 2021a) and multi-agent systems (Gao et al. 2016).

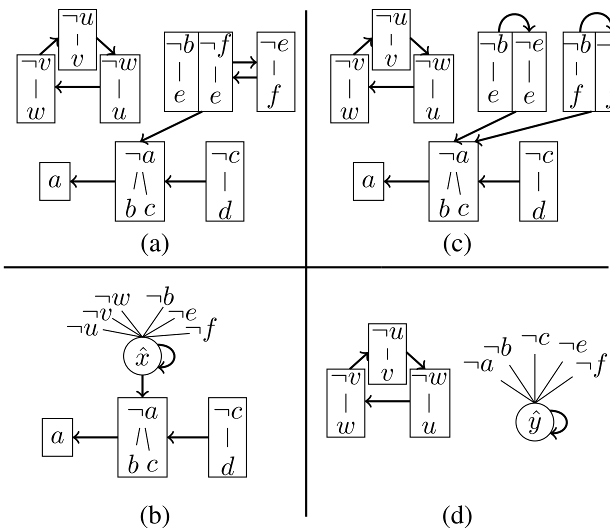


Figure 1: In this figure we show two ABAFs (top) and two frameworks (bottom) where the latter occur after applying a clustering on the former. Assumptions attack each other based on what atoms they derive. Different conclusions by the same assumption are concatenated.

Example 1. *Let us illustrate our approach with an example. All details will be formalized in the main body of the paper. For now, let us consider an assumption-based argumentation framework. Suppose we are thinking of purchasing our own house. However, being unemployed and having no savings prevents us from doing so, as under these assumptions, we have no money. One way to get a house, would be to get a job. Another would be to get a loan. Getting a loan, albeit being a valid solution that would grant us enough money, is not the best idea in general. This is a possible interpretation to Figure 1(a), where* a *stands for our need for a house,* b *and* c *for the assumption of having no savings and of being unemployed respectively. Then* d *represents the assumption of a job offering,* e *is the possibility of getting a loan, and lastly* f *signals that getting a loan is not a good idea. On the other hand, independently of our desire for a new home, we are also bothered by our colleagues, who accuse each other of being liars (example by Bench-Capon, 2016). In fact,* w *claims that* v *is lying,* v *does not trust* u*, and eventually* u *does not trust* w*.*

Argumentation semantics define which sets of assumptions can be deemed acceptable. A main ingredient is that of admissibility of a set of assumptions. A set of assumptions is conflict-free if one cannot derive a contrary of the set. A conflict-free set A *is admissible if all assumptions sets* B *that attack* A *(i.e., derive a contrary within* A*) are attacked by* A*, i.e.,* A *defends itself.*

For instance, in the example framework, there are admissible sets containing assumption a *and giving an argumentative reason for accepting* a*, e.g.,* $\{a, d\}$ *and* $\{a, e\}$*. Consider now a "clustering" of the assumptions* e*,* f*,* u*,* v*, and* w *into one cluster that we call* $\hat{x}$ *(Figure 1(b)). In Figure 1(b), attacks from clusters are interpreted existentially, e.g., there is an attack from within the cluster onto* b*. In such a clustered ABA framework the sets* $\{a, d\}$ *and* $\{a, \hat{x}\}$ *are admissible, with the formalization we provide below. For intuitions, the set* $\{a, d\}$ *is unchanged, and the set* $\{a, \hat{x}\}$ *states that* a *needs a defender (against* b*) and there is a way to attack* b*, and defend* a*, from within the cluster, but the details have been abstracted away. Considering the meaning of the symbols,* $\{a, d\}$ *can be interpreted as buying a house can be defended when we start a new job. The set* $\{a, \hat{x}\}$ *can be interpreted existentially: buying a house can be argumentatively defended by some assumptions in the cluster* $\hat{x}$*, but which one(s) was abstracted away. In particular, also all unrelated assumptions (*u*,* v*, and* w*) were abstracted away. In our methodology below, we also allow for interactive querying of a user to "zoom in" into a cluster to get more information, and, e.g., to extract assumptions from the cluster.*

Nevertheless, not any clustering, or any abstraction, gives rise to sound reasoning, as in this case. For instance, Figure 1(c) shows a different ABA framework that, when applying the same clustering (e*,* f*,* u*,* v*, and* w *into* $\hat{x}$*), results in the same clustered framework of Figure 1(b). That is, after clustering, these two ABA frameworks coincide. However, in case of the ABA framework in Figure 1(c), there is no admissible set corresponding to* $\{a, \hat{x}\}$ *(defense of* a *requires* d*). Hence, this set is "spurious" under admissibility, since it allows to draw a conclusion not warranted on the original ABA instance.*

As a different direction, consider another prominent semantics of ABA: stable semantics. There is no stable set in both the example ABA frameworks, intuitively due to the odd-cycle between assumptions u*,* v*, and* w*. The same behaviour can be seen when clustering all assumptions, except these three, into one big cluster, thereby abstracting away information not needed to see the behaviour of having no stable assumption sets, see Figure 1(d). This matches also the intuition behind the example: the accusations and the issue of buying a house are unrelated, and the non-existence of a stable assumption set relies on the odd-cycle.*

Main contributions of our work are as follows.

- We introduce clusterings on assumptions in ABA frameworks, applying a form of existential abstraction.
- We present semantics of the resulting clustered ABA frameworks (cABAFs) for conflict-free sets, admissible sets, and stable sets, i.e., the counterparts of classical ABA semantics when a clustering is taken into account, and show several properties. We define spuriousness and faithfulness, i.e., non-spurious reasoning on cABAFs. We show that the novel semantics do not exhibit an increase in complexity w.r.t. classical ABA semantics.
- We show that clusterings (abstractions) on AFs and ABAs are not only distinct, but the latter can produce less spuriousness and is also proved to be more involved. In addition to including different types of strength of attacks in cABAFs, we show that defining semantics in an optimal way to prevent spuriousness faces an additional complexity barrier compared to such semantics of AFs.

- For the semantics we propose, while it is in general complex to find faithful clusterings, we present two ways of obtaining a clustering and an interactive tool based on answer set programming (ASP) (Brewka, Eiter, and Truszczyński 2011) for one of them. Both are capable of obtaining faithful abstractions.
- Using the tool we present use cases for abstracting large parts of a given ABA and focusing (i) on reasons for not having stable sets and (ii) on interactions of assumptions towards acceptance under admissibility.

The prototype and more details are available online.[1]

2 Assumption-based Argumentation

We recall assumption-based argumentation (ABA) (Bondarenko et al. 1997; Čyras et al. 2018). One main part of ABA is a so-called deductive system $(\mathcal{L}, \mathcal{R})$, with $\mathcal{L}$ a formal language and $\mathcal{R}$ a set of inference rules over $\mathcal{L}$. In this work we assume that $\mathcal{L}$ is a set of atoms. A rule $r \in \mathcal{R}$ is of the form $a_0 \leftarrow a_1, \ldots, a_n$ with $a_i \in \mathcal{L}$. We define a shorthand for the head of a rule r by $head(r) = a_0$, and for the (possibly empty) body via $body(r) = \{a_1, \ldots, a_n\}$. An ABA framework (ABAF) contains a deductive system and specifies which atoms are assumptions and their contraries.

Definition 1. *An ABAF is a tuple $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, where $(\mathcal{L}, \mathcal{R})$ is a deductive system, $\mathcal{A} \subseteq \mathcal{L}$ a non-empty set of assumptions, and $^-$ a total function, mapping assumptions $a \in \mathcal{A}$ to atoms $s \in \mathcal{L}$ (the contrary function).*

We extend the contrary function to sets: $\overline{S} = \{\overline{x} \mid x \in S\}$. In this work, we focus on ABAFs which are flat, i.e., no assumption can be derived: we require for each rule $r \in \mathcal{R}$ that $head(r) \notin \mathcal{A}$ holds. We assume that ABA frameworks are finite ($\mathcal{L}$, $\mathcal{R}$, and $body(r)$ for each $r \in R$ are finite). We also make the convention of having a single contrary for each assumption instead of multiple ones. This is a choice that is made mostly for presentational and accessibility reasons. In general, one can modify an ABAF where each assumption has multiple contraries, to acquire a modified framework with single contraries (Toni 2014).

Semantics of ABAFs can be defined using subsets of assumptions and using derivation trees (proof trees), with the latter interpreted as arguments. We focus on the assumption-based definition. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF. An atom $s \in \mathcal{L}$ is derivable from a set of assumptions $A \subseteq \mathcal{A}$ if $s \in A$ or there is a sequence of rules $(r_1, \ldots, r_n)$ such that $head(r_n) = s$ and for each rule $r_i \in \mathcal{R}$ we find that its body is derived from earlier rules in the sequence or is in A, i.e., $body(r_i) \subseteq A \cup \bigcup_{j<i}\{head(r_j)\}$. Then, $Th_D(A)$ contains all atoms derivable from A in D.

For illustration, we sometimes present derivations as rooted proof trees (arguments), with assumptions in leaves, internal nodes corresponding to rule derivations, and the root showing a derived atom.

A set A of assumptions attacks a set B of assumptions if one can derive the contrary of an assumption in B from A.

[1] http://www.kr.tuwien.ac.at/research/systems/abstraction/

Definition 2. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF and $A, B \subseteq \mathcal{A}$ be two sets of assumptions. Assumption set A attacks assumption set B in D, if $\overline{b} \in Th_D(A)$ for some $b \in B$.*

When A attacks $\{b\}$, we sometimes omit the brackets.

Definition 3. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF. An assumption set $A \subseteq \mathcal{A}$ is conflict-free in D, $A \in cf(D)$, iff A does not attack itself. Moreover, A defends assumption set $B \subseteq \mathcal{A}$ in D iff for all $C \subseteq \mathcal{A}$ that attack B it holds that A attacks C.*

Main semantics of ABA can now be defined, as follows.

Definition 4. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF and let $A \in cf(D)$. Then the set A is*

- admissible*, $A \in adm(D)$, iff A defends itself, and*
- stable*, $A \in stb(D)$, iff A attacks each $\{x\} \subseteq \mathcal{A} \setminus A$.*

In this paper we focus on the semantics defined above, as for other semantics (e.g. complete) we signal through an example, that abstracting them is more involved.

For reasoning, an atom s is credulously accepted under σ in an ABAF D iff there is a σ-assumption-set A s.t. $s \in Th_D(A)$.

Example 2. *As a running example we consider the ABAF D shown in Figure 2(a) with assumption set $\mathcal{A} = \{a, b, c, d, e\}$, $\mathcal{L} = \{a, b, c, d, e, x, y, z, w, t\}$, and contraries, $\overline{a} = z$, $\overline{b} = t$, $\overline{c} = x$, $\overline{d} = w$, $\overline{e} = t$. Consider also the rules of this framework to be the following:*

$$x \leftarrow a, b, \qquad w \leftarrow e,$$
$$y \leftarrow c, \text{ and } \qquad z \leftarrow d.$$

In the figure, we omitted the arguments consisting solely of assumptions (proof trees concluding assumptions), for the sake of a concise presentation. We remark that including them would make no difference to our results. According to the definition above, the set $\{b, c, e\}$ is conflict-free, and this set is also stable and admissible. On the contrary the sets $\{a, b, c\}$ and $\{a, b, d\}$ are not conflict-free, and thus they are neither admissible nor stable.

3 Clustering Assumptions

Let us start by defining the notion of existential abstraction in ABAFs. In this work, abstraction refers to clustering the set of assumptions of an ABAF, obtained by using a mapping m, intuitively mapping assumptions to clusters. Formally, we can view the mapping m as a partition of the original set of assumptions. For instance, for $\mathcal{A} = \{a, b, c\}$ we can map a and b to the same cluster, i.e., clustering them together, and leave c unclustered. An associated mapping m would be $m(a) = m(b) = \hat{a}$ and $m(c) = c$, with the "hat" notation $\hat{a}$ denoting a cluster. Assumption c is mapped to itself (signaling no clustering). Equivalently, we can partition $\mathcal{A} = \{a, b, c\}$ by $\hat{\mathcal{A}} = \{\{a, b\}, \{c\}\}$. Assumption c is in a singleton set, and we refer to such singleton "clusters" also directly as singletons. For simplicity, we use partitions and mappings sometimes interchangeably (e.g., viewing clusters $\hat{a}$ as sets containing their assumptions).

We extend m to sets of assumptions straightforwardly: $m(A) = \{\hat{a} \mid m(a) = \hat{a}, a \in A\}$.

A clustering of assumptions leads to rules and the contrary function being modified accordingly.

Definition 5. *Given an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$, let $\hat{\mathcal{A}}$ be a partition of $\mathcal{A}$ and m be the surjective mapping from $\mathcal{A}$ to $\hat{\mathcal{A}}$. Then, the clustered ABAF (cABAF) of D according to m is $\hat{D} = m(D) = (\hat{\mathcal{L}}, \hat{\mathcal{R}}, \hat{\mathcal{A}}, \hat{\overline{}})$ where*

- $\hat{\mathcal{L}} \setminus \hat{\mathcal{A}} = \mathcal{L} \setminus \mathcal{A}$ *and* $\hat{\mathcal{A}} = m(\mathcal{A})$,
- *each rule $r \in \mathcal{R}$ is mapped to the rule $\hat{r}$ of the form $head(r) \leftarrow m(body(r))$, hence obtaining $\hat{\mathcal{R}}$,*
- $\hat{\overline{}}$ *is a total mapping from $\hat{\mathcal{A}}$ to $2^{\hat{\mathcal{L}}}$ such that for all $\hat{a} \in \hat{\mathcal{A}}$ we have $\hat{\overline{\hat{a}}} = \{\hat{b} \in \hat{\mathcal{A}} \mid \exists b \in \hat{b}, a \in \hat{a} \text{ s.t. } b = \overline{a}\} \cup \{\hat{x} \in \hat{\mathcal{L}} \setminus \hat{\mathcal{A}} \mid \exists a \in \hat{a} \text{ s.t. } x = \overline{a}\}$.*

When there is no danger of ambiguity regarding the contrary of a cluster, we will omit the double-hat notation.

We remark that a clustered ABAF is not a "classical" ABAF. The contrary function of a classical ABAF maps elements of its assumption set to a particular atom in its formal language. Instead, the contrary function of a clustered ABAF maps clusters to subsets of the clustered formal language. When applying a mapping to a classical ABAF, contraries of clusters can map either to (clustered) assumptions or non-assumptions, leading to a case distinction in the definition above.

Example 3. *Remember the framework described in Example 2, and let m be the mapping that maps assumptions d and c into the same cluster $\hat{c}$, and the assumptions a, b, e to themselves. This mapping results in cABAF $\hat{D}$ (see Figure 2(b)), whose assumption set is $\hat{\mathcal{A}} = \{a, b, \hat{c}, e\}$. As mentioned previously, the contrary function of a cABAF does not fit in the requirements of the definition of a classical ABAF. This is evident in this example, since the contrary of the cluster $\hat{c}$ is not an element of the set $\hat{\mathcal{L}} = \{a, b, \hat{c}, e, x, y, z, w, t\}$. In fact, $\overline{\hat{c}} = \{x, w\} \notin \hat{\mathcal{L}}$.*

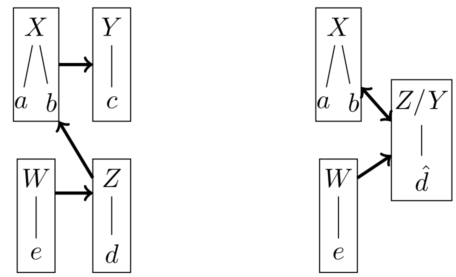
Regarding the rules, as Definition 5 states, for each rule in $\mathcal{R}$, we get a clustered rule by applying m to its body. Therefore, the clustered rule set $\hat{\mathcal{R}}$ contains the following rules: $(x \leftarrow a, b)$, $(w \leftarrow e)$, $(y \leftarrow \hat{c})$, and $(z \leftarrow \hat{c})$.

Abstraction on flat ABAFs leads to flat clustered ABAFs. Derivation in cABAFs is defined as for ABAFs.

Derivability of a classical ABAF and a clustering of this ABAF is connected, as stated next. We define $Single(\hat{A}) = \{\hat{a} \in \hat{A} \mid |\hat{a}| = 1\}$ to be the set of singletons in $\hat{A}$.

Proposition 1. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF, m a mapping, and $m(D) = \hat{D} = (\hat{\mathcal{L}}, \hat{\mathcal{R}}, \hat{\mathcal{A}}, \hat{\overline{}})$ the resulting cABAF. Moreover, let $S \subseteq \mathcal{A}$ and $m(S) = \hat{S} \subseteq \hat{\mathcal{A}}$.*

1. *It holds that $Th_D(S) \subseteq Th_{\hat{D}}(\hat{S})$. However, when $x \in Th_{\hat{D}}(\hat{S})$ it is not necessarily true that x is in $Th_D(S)$.*
2. *If we additionally assume that $\hat{S} \subseteq Single(\hat{\mathcal{A}})$, then $Th_{\hat{D}}(\hat{S}) = Th_D(S)$.*
3. *In general, $Th_{\hat{D}}(Single(\hat{S})) \subseteq Th_D(S)$.*
4. *It holds that $Th_D(\mathcal{A}) = Th_{\hat{D}}(\hat{\mathcal{A}})$.*
5. *If $x \in Th_{\hat{D}}(\hat{S})$, then there is $S' \subseteq \mathcal{A}$ s.t. $x \in Th_D(S')$.*



(a) (b)

Figure 2: Classical ABAF (a) and clustered ABAF (b) in an argument-based view. On the right, cluster $\hat{d}$ derives both Z and Y, thus we represent these derivations together as one argument.

The first three statements give us a hint on the difference of deriving an atom by singletons, in comparison to deriving it by clusters. Singletons preserve the information of the original ABAF, and this is why we treat them in a special way. The formula in (4) shows that all derivable atoms in D, are also derivable in $\hat{D}$, while (5) states that if a clustered set derives an atom, then there must be some $S' \subseteq \mathcal{A}$ that derives this atom too.

In this work, when given a clustered ABAF $\hat{D}$, we assume that $\hat{D}$ has an associated (fixed) partition function m. We sometimes view cABAFs $\hat{D}$ without concrete ABAFs D s.t. $m(D) = \hat{D}$. In such instances, a mapping is nevertheless assumed to be known. In other words, we assume knowledge of contents of each cluster. This implies that the underlying non-clustered assumption set is given by $\mathcal{A} = m^{-1}(\hat{\mathcal{A}})$.

It turns out that it is useful to distinguish three kinds of attacks on cABAFs, each representing different "strengths" of attacks, when considering the clustering.

Definition 6. *Let $\hat{D} = (\hat{\mathcal{L}}, \hat{\mathcal{R}}, \hat{\mathcal{A}}, \hat{\overline{}})$ be a cABAF, and $\hat{A}, \hat{B} \subseteq \hat{\mathcal{A}}$. We say that*

- *$\hat{A}$ (normally) attacks $\hat{B}$ if $\exists \hat{b} \in \hat{B}$ and $\overline{\hat{b}} \cap Th_{\hat{D}}(\hat{A}) \neq \emptyset$,*
- *$\hat{A}$ fully attacks $\hat{B}$ if $\exists \hat{b} \in \hat{B}$ and $\overline{\hat{b}} \subseteq Th_{\hat{D}}(\hat{A})$, and*
- *$\hat{A}$ truly attacks $\hat{B}$ if $\exists \hat{b} \in \hat{B}$ and $\overline{\hat{b}} \subseteq Th_{\hat{D}}(Single(\hat{A}))$.*

When $\hat{B} = \{\hat{b}\}$, then we omit the curly brackets and say that $\hat{A}$ (normally, fully, truly) attacks $\hat{b}$. In words, a set of clustered assumptions $\hat{A}$ (possibly containing singletons) attacks $\hat{b}$ normally similarly as in classical ABAFs by deriving some contrary of $\hat{b}$ (except here there can be multiple contraries). A full attack represents the case that all contraries are derived. That is, each original assumption in the cluster is attacked, even after clustering. True attacks, the strongest kind, additionally require that singletons among $\hat{A}$ fully attack, implying that non-abstracted assumptions carry out the full attack.

Example 4. *Continuing Example 3, a (normal) attack occurs when a set of clustered assumptions can derive an atom that is a contrary of another cluster. When this happens we say that the set attacks the latter cluster. Using the example, we can see that the set $\{a, b\}$ attacks cluster $\hat{c}$. That is because the set $\{a, b\}$ can derive the atom x, and also it holds that $\overline{\overline{c}} = \{x, w\}$, which entails that $\overline{\overline{a}} \cap Th_{\hat{D}}(\{a, b\}) \neq \emptyset$.*

An aspect of this type of attack is that clusterings may introduce new (normal) attacks not part of the classical one. For instance, in Figure 2, assumption set $\{e\}$ is not attacking any of the premises of a rule that derives atom y. However, this changes after the clustering: $\{e\}$ attacks $\hat{c}$, which is now in the body of a rule that derives y. This subtle detail is important for defining the notion of defense in cABAFs.

We say that a set of clusters fully attacks a cluster, if this set can derive all the contraries of the cluster. In Figure 2(b), the singleton cluster a has exactly one contrary, namely z. Cluster $\hat{c}$ derives atom x, landing a full attack on a.

Finally, a true attack is a special case of a full attack, for we now do care about the size of the clusters. More specifically, a true attack happens when a set of singletons derives all the contraries of a cluster. An example of such an attack can be viewed in Figure 2(b) between the set of singletons $\{a, b, e\}$ and the cluster $\hat{c}$.

4 Semantics of Clusterings

In this section we define semantics of cABAFs, with the aim of abstracting classical semantics of ABAFs, by introducing as little spuriousness as possible.

A semantics $\hat{\sigma}$ in $\hat{D}$ is a set $\hat{\sigma}(\hat{D}) \subseteq 2^{\hat{\mathcal{A}}}$. To distinguish between semantics on ABAFs and semantics on cABAFs we will, whenever not obvious from the context, refer to the former as classical and the latter as abstract semantics.

We now define main concepts of abstract semantics $\hat{\sigma}$ and their relation to a classical semantics σ. For the assumption sets under a semantics, i.e., set of sets of assumptions $\sigma(D)$, we define $m(\sigma(D)) = \{m(A) \mid A \in \sigma(D)\}$.

Definition 7. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF and $\hat{D}$ be its clustered framework according to some partition function m. We say that $\hat{D}$ under $\hat{\sigma}$*

- *abstracts D under σ if $m(\sigma(D)) \subseteq \hat{\sigma}(\hat{D})$ and*
- *is faithful w.r.t. D under σ if $m(\sigma(D)) = \hat{\sigma}(\hat{D})$.*

Moreover, for $\hat{A} \in \hat{\sigma}(\hat{D}) \setminus m(\sigma(D))$, we say that $\hat{A}$ is spurious (in $\hat{D}$ under $\hat{\sigma}$ w.r.t. D under σ). If such a set exists, then $\hat{D}$ under $\hat{\sigma}$ is spurious w.r.t. D under σ.

Intuitively, an abstract semantics is abstracting a classical semantics if the abstract semantics preserves, under the mapping, the assumption sets under the classical semantics. That is, no σ-assumption-set is missed. Faithfulness means that the abstract semantics is abstracting and does not introduce spuriousness, in the form of $\hat{A} \in \hat{\sigma}(\hat{D})$ which have no counterpart in the concrete $\sigma(D)$.

We are now ready to define our abstract semantics for conflict-freeness, admissibility, and stable semantics. We begin with the notion of defense.

Definition 8. *A set of clusters $\hat{A}$, defends a cluster $\hat{a}$ that is fully attacked in $\hat{D}$, if there is an atom $x \in \overline{\overline{a}}$ such that for all sets of clusters $\hat{C}$ that derive x, $\hat{A}$ attacks a cluster in $\hat{C}$. Additionally, we consider clusters that are not fully attacked in $\hat{D}$ to be defended, by any set.*

If a cluster is fully attacked, this means that all of the elements in the cluster receive an attack in any classical framework that maps to this setting. Defending against any contrary of the cluster, means that there is an element of the cluster that is actually defended in some classical framework by a preimage of the set of clusters.

Definition 9. *Let $\hat{D} = (\hat{\mathcal{L}}, \hat{\mathcal{R}}, \hat{\mathcal{A}}, \hat{\overline{}})$ be a cABAF. A set of clusters $\hat{A} \subseteq \hat{\mathcal{A}}$ is*

- *conflict-free in $\hat{D}$ if it does not attack itself truly,*
- *admissible, if it is conflict-free and it defends all of its clusters, and*
- *stable iff it is conflict-free, $\forall \hat{a} \notin \hat{A}$ there must be a full attack from $\hat{A}$ to $\hat{a}$, and if $\hat{a} \in \hat{A}$, then if $\hat{S} \subseteq \hat{A}$ is a set of clusters that fully attack $\hat{a}$, then $\hat{A}$ must attack at least one of the clusters in $\hat{S}$.*

In other words, for a conflict-free set it holds that $\hat{A} \in \hat{cf}(\hat{D})$ iff $\forall \hat{a} \in \hat{A}$, $\overline{\overline{a}} \nsubseteq Th_{\hat{D}}(Single(\hat{A}))$. Intuitively, if all of the vulnerabilities of a cluster (i.e., atoms that are contraries to an element of the cluster) are derived by singletons of the set, then there can be no concrete framework in which a preimage of this set does not contain any attacks. The point of Definition 9 is that only if there are no true attacks involved in set of clustered assumptions, can we guarantee that for some concrete framework, this clustered set is not spurious.

An admissible set of clusters might receive many attacks, but in no case can it be that a cluster has all its elements undefended. Alternatively, $\hat{A} \in \hat{adm}(\hat{D})$ if the set $\hat{\mathcal{A}} \setminus att(\hat{A})$ does not fully attack $\hat{A}$, where $att(\hat{A})$ is the set of clusters that are attacked by $\hat{A}$.

In the case of stable assumption sets, the definition differs a bit more from the classical counterpart. This is mainly because of the last requirement, which eliminates sets containing fully attacked clusters by unattacked members of the set. The reasoning behind this requirement is that a cluster of a stable assumption set can be fully attacked by the assumption set itself and still not be spurious. However, if a cluster is contained and fully attacked by unattacked members of the set, there is no way that there is a classical stable assumption set that maps to this set, since then it is not conflict-free. Thus, adding this property eliminates those extra spurious sets that contain such attacks.

As in classical ABAFs, a stable set is also admissible.

Proposition 2. *Let $\hat{D}$ be a cABAF. It holds that each abstract stable assumption set in $\hat{D}$ is abstract admissible in $\hat{D}$.*

Example 5. *Let us continue Example 4. The set of singletons $\{a, b, e\}$ truly attacks the cluster $\hat{c}$. Hence, the set $\{a, b, e, \hat{c}\}$ is not conflict-free. On the contrary, the set $\{b, e, \hat{c}\}$ is conflict-free in $\hat{D}$. Now to see whether this set* *is spurious, we only need to check if any of its preimages in D is conflict-free under the classical semantics. Indeed, the set $\{b, e, c\}$ is conflict-free in D, and also $m(\{b, e, c\}) = \{b, e, \hat{c}\}$. Hence, this set is not spurious. However this clustering is not faithful under conflict-free semantics, and this becomes clear when looking at the set $\{a, b, \hat{c}\}$. This set is spurious, as it is conflict-free in $\hat{D}$ since it contains no true attacks, but none of its preimages, i.e., $\{a, b, d\}$, $\{a, b, c, d\}$, and $\{a, b, c\}$ are conflict-free in D. However, we cannot avoid such spuriousness, as there is an ABAF D' with $m(D') = \hat{D}$ such that $\{a, b, \hat{c}\}$ is conflict-free in D'. As a matter of fact, in Figure 3 we present an ABAF D' which maps to $\hat{D}$ under the same clustering and in which the same set is not spurious. We formalize the framework in Figure 3(a) as follows:*

- $\mathcal{A} = \{a, b, c, d, e\}$, $\mathcal{L} = \{a, b, c, d, e, x, y, z, w, t\}$,
- $\overline{a} = z$, $\overline{b} = t$, $\overline{c} = w$, $\overline{d} = x$, $\overline{e} = t$,
- $\mathcal{R}' = \{x \leftarrow a, b, w \leftarrow e, y \leftarrow c, z \leftarrow d\}$.

Applying the same clustering on D', i.e., $m(a) = a, m(b) = b, m(c) = m(d) = \hat{d}, m(e) = e$, results in the same cABAF $\hat{D}$, only now the set $\{a, b, c\}$ is conflict-free in D. Thus in this case, the set $\{a, b, \hat{d}\}$ is not spurious.

In Example 4 cluster $\hat{c}$ attacks fully the singleton a. This implies that $\{a, \hat{c}\} \notin \hat{adm}(\hat{D})$. However, the singleton e defends a from atom z, and consequently $\{a, e\} \in \hat{adm}(\hat{D})$. At this point, we just remind that attacks can occur in admissible set, e.g., $\{a, e, \hat{c}\} \in \hat{adm}(\hat{D})$.

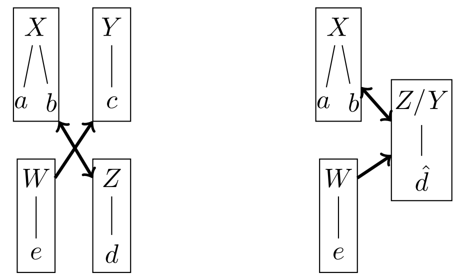
The sets $\{b, e, a\}$ and $\{b, e, \hat{c}\}$ are both stable under the abstract semantics. The former set is a set of singletons, and one can directly check that it is also stable in the classical framework. On the other hand, the latter is spurious, as none of the sets $\{b, e, d\}$, $\{b, e, c\}$, and $\{b, e, c, d, \}$ is stable.

As a direct property, if a clustered ABAF has only singleton assumptions (no abstraction is applied), then the abstract semantics coincide with the classical semantics (additionally we also require that the contrary function maps only to one contrary per assumption as in classical ABAFs). One observation is that when all clusters are singletons any attack between them is true.

Proposition 3. *Let $\hat{D} = (\hat{\mathcal{L}}, \hat{\mathcal{R}}, \hat{\mathcal{A}}, \hat{\ })$ be a cABAF for which it holds that $Single(\hat{\mathcal{A}}) = \hat{\mathcal{A}}$ and $\hat{\ }$ assigns only single atoms to assumptions. Then classical conflict-free, admissible, and stable semantics coincide with abstract conflict-free, admissible, and stable semantics.*

Extending the definition of abstracting to semantics, we say that a semantics $\hat{\sigma}$ on cABAFs abstracts σ on ABAFs if for each ABAF D and mapping m with $m(D) = \hat{D}$ it holds that $\hat{D}$ under $\hat{\sigma}$ abstracts D under σ. We show that our abstract semantics are all abstracting their classical counterparts.

Theorem 1. *It holds that the abstract conflict-free, admissible, and stable semantics abstract classical conflict-free, admissible, and stable semantics, respectively.*



(a) (b)

Figure 3: Framework (a) is a preimage of (b), w.r.t. which $\{a, b, \hat{c}\}$ is not spurious under conflict-freeness.

Proof for conflict-free. Let D be an ABAF and $m(D) = \hat{D}$ a cABAF. Let $A \subseteq \mathcal{A}$ be a conflict-free set, such that $m(A) = \hat{A} \notin \hat{cf}(\hat{D})$. Then $\exists \hat{a} \in \hat{A}$ s.t. $\overline{\hat{a}} \subseteq Th_{\hat{D}}(Single(\hat{A}))$. However, $Th_{\hat{D}}(Single(\hat{A})) \subseteq Th_D(A)$. Hence, $\forall a \in A$ with $m(a) = \hat{a}, \exists B \subseteq A$, s.t. B attacks a. This contradicts the assumption that $A \in cf(D)$. $\square$

This is a key property of the abstract semantics: it is guaranteed that through abstracting we do not lose any essential information regarding assumption sets. However, we still have to avoid the threat of excessive spuriousness. Spuriousness cannot be avoided completely. We can, however, aim to minimize it as much as possible. The following result shows that the definition of conflict-freeness for cABAFs is optimal w.r.t. spuriousness.

Theorem 2. *Let $\hat{D} = (\hat{\mathcal{L}}, \hat{\mathcal{R}}, \hat{\mathcal{A}}, ^-)$ be a clustered ABAF according to a mapping m. Let also $\hat{\tau}$ be a clustered semantics that abstracts cf. Then we have $\hat{cf}(\hat{D}) \subseteq \hat{\tau}(\hat{D})$.*

Sketch proof. Aiming to a contradiction, we assume that there is a clustered ABAF $\hat{D}$ s.t. $\hat{cf}(\hat{D}) \subseteq \hat{\tau}(\hat{D})$ does not hold, i.e., $\exists \hat{A} \in \hat{cf}(\hat{D})$ s.t. $\hat{A} \notin \hat{\tau}(\hat{D})$. We will use that to construct a classical ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ such that, (i) $m(D) = \hat{D}$ and (ii) $m(A) = \hat{A}$ for some $A \in cf(D)$. In our construction, $\mathcal{L} = m^{-1}(\hat{\mathcal{L}})$ and $\mathcal{A} = m^{-1}(\hat{\mathcal{A}})$. The set $A = \bigcup_{\hat{a} \in \hat{A}} \{c_a\}$ contains one representative from each cluster in $\hat{A}$, and the rules and contrary relation are defined in a way to avoid derivations and any conflict within this set. $\square$

In contrast to conflict-free semantics, finding optimal admissible semantics for cABAFs seems to be more involved. We show this formally, by utilizing a complexity-theoretic result. The next theorem states that the complexity of deciding, given a specific cABAF and a clustered assumption set, whether this set is spurious under all possible preimages of this cABAF is computationally involved.

Theorem 3. *Deciding if there exists an ABAF and an admissible set of assumptions mapping to a given cABAF and given set of clustered assumptions, respectively, is NP-hard.*

Sketch proof. We reduce from SAT. Let $\phi = c_1 \wedge \cdots \wedge c_m$ be a formula in CNF over variables X. Construct $\hat{D} = (\hat{\mathcal{L}}, \hat{\mathcal{R}}, \hat{\mathcal{A}}, ^-)$, with $\hat{\mathcal{A}} = \bigcup_{x \in X}\{\hat{y_x}\} \cup \bigcup_{i \leq m}\{d_i, f_i\} \cup \{e\}$, where $\hat{y_x} = \{y_{x_1}, y_{x_2}\}$ are clusters with two concrete assumptions and d_i,f_i for $i \leq m$ and e singletons. Define $\hat{\mathcal{R}} = \bigcup_{x \in X}\{\hat{y_x}\} \cup \bigcup_{i \leq m}\{d_i, f_i\} \cup \{e\}\{(x \leftarrow \hat{y_x}), (\neg x \leftarrow \hat{y_x}), (z \leftarrow x, \neg x)\}$, and the contrary relations: $c_i = \overline{d_i}$, $d_i = \overline{f_i}$, $z = \overline{e}$, $e = \overline{\hat{y_x}}$, where each c_i is an atom representing each clause of ϕ. Finally we define $\hat{A} = \bigcup_{x \in X}\{\hat{y_x}\} \cup \bigcup_{i \leq m}\{f_i\} \cup \{e\}$.

If ϕ is satisfiable, let $Y \subseteq X$ be a model of ϕ. Construct rule set $\mathcal{R}$ of D by $x \leftarrow y_{x_1}$ and $\neg x \leftarrow y_{x_2}$ if $x \in Y$, $x \leftarrow y_{x_2}$ and $\neg x \leftarrow y_{x_1}$ if $x \notin Y, (z \leftarrow x, \neg x), c \leftarrow x$, if x appears in c. For contraries in D let $e = \overline{y_{x_1}}$, if $x \notin Y e = \overline{y_{x_2}}$, if $x \in Y$. Set $A = \{y_{x_1}, x \in Y\} \cup \{y_{x_2}, x \notin Y\} \cup \{e\} \cup \{f_i\}$, maps to $\hat{A}$ and is admissible. If ϕ is unsatisfiable, let A, D s.t. the claim holds. Set A cannot contain both y_{x_1} and y_{x_2} for $x \in X$. If A contains y_{x_1} (y_{x_2}), then $\overline{y_{x_2}} = e$ ($\overline{y_{x_1}} = e$). Since ϕ is unsatisfiable not enough of $c \leftarrow x$ can be triggered, which in turn means that an A that maps to $\hat{A}$ cannot be admissible. □

In more words, this result states that, if one would define a semantics $\hat{\tau}$ that "optimally" defines an abstracting counterpart to classical admissibility requires conditions that are NP-hard to check. That is, if one defines $\hat{\tau}$ to exactly contain only those clustered assumption sets $\hat{A}$, for given $\hat{D}$ and m, for which some classical ABAF D with $m(D) = \hat{D}$ exists who has a matching A ($m(A) = \hat{A}$) requires conditions to check whether $\hat{A} \in \hat{\tau}(\hat{D})$ that are NP-hard.

On the other hand, the semantics we defined (Definition 9) exhibit the same complexity as the corresponding semantics in classical ABA frameworks.

Proposition 4. *One can in polynomial time decide whether a given set of clustered assumptions is conflict-free, admissible, or stable in a given cABAF. It is NP-complete to decide credulous acceptance under admissibility or stable semantics in cABAFs.*

We remark that our complexity statements are not contradictory: computation of abstract semantics of a given cABAF has the same complexity as in classical ABAFs. However, an *optimal* variant of admissibility is more involved, i.e., it is NP-hard to verify whether a set of clustered assumption is part of such an optimal admissible semantics. These complexity results also set apart existential abstraction in AFs (Saribatur and Wallner 2021) and existential abstraction in cABAFs: in AFs optimal admissibility does not exhibit a "complexity jump". Due to this complexity jump, we opted to focus on our notion of abstract admissibility (with the same complexity as classical ABA semantics). See also Section 6 for more relations to AFs.

Complexity is also reflected when checking whether a clustered assumption set is spurious, given an ABAF and cABAF.

Proposition 5. *Given an ABAF D and its clustered ABAF $\hat{D}$, w.r.t. the clustering m, deciding whether a clustered assumption set is spurious under admissibility is coNP-complete.*

The preceding proposition suggests that finding clusterings is challenging, in general: when searching for a faithful cABAF, one needs to (also) solve non-spuriousness. We provide two ways of finding faithful cABAFs in Section 5.

Up to now, we focused on conflict-free, admissible, and stable semantics. However, what happens when we consider other semantics like complete, preferred and grounded? These semantics seem to be more involved. Defining an abstract semantics that abstract complete semantics is not impossible, since already the abstract admissible semantics are indeed abstracting the former. However, this abstraction is of low importance as it barely progresses towards what would be an optimal abstraction. Let us briefly remind the reader of the complete, preferred and grounded semantics. A complete set of assumptions is an admissible set that contains all the assumptions it defends. A preferred set is a subset-maximal complete set, while the grounded set is the minimal complete set. Initially, one could attempt to define an abstract complete semantics in a straightforward way, by finding sets that contain all the clusters they defend, imitating the classical complete mechanism. In the following example we see that such a naive approach, although it seems to be in the right direction, does not bear any fruit.

Example 6. *Consider the following ABAF:*

$$\mathcal{L} = \{\neg c, \neg b\} \cup \mathcal{A},\ \mathcal{A} = \{a, b, c, c'\},$$
$$\mathcal{R} = \{\neg c \leftarrow a, \neg b \leftarrow c'\},\ \overline{b} = \neg b, \overline{c} = \neg c.$$

In this framework the sets $\emptyset, \{a\}, \{c'\}, \{a, c'\}$ are admissible. Out of these sets, only the set $\{a, c'\}$ is complete.

Let us now consider a clustering on this framework through mapping m, where $m(a) = a, m(b) = b, m(c) = m(c') = \hat{c}$. In the resulting cABAF we find that the admissible sets are $\emptyset, \{a\}, \{\hat{c}\}, \{a, \hat{c}\}, \{a, b\}, \{a, b, \hat{c}\}$. To define abstract complete sets, let us make use of the function $\mathcal{F}_{\hat{D}}(\hat{S}) = \{\hat{a} \in \hat{\mathcal{A}} \,|\, \hat{a}$ is defended by $\hat{S}\}$. If we were to consider as abstract complete sets all the sets that are fixed points of $\mathcal{F}$ then the set $\{a, b, \hat{c}\}$ would be the only fixed point. Here, we observe that there is a mismatch between the complete set of D and the abstract complete set of $\hat{D}$, since $\{a, c'\} \in com(D)$, but $m(\{a, c'\}) = \{a, \hat{c}\} \notin \hat{com}(\hat{D})$.

Additionally, using the same example we can draw some conclusions regarding preferred and grounded semantics. In the ABAF above the set $\{a, c'\}$ is also the grounded and preferred set, since it the only complete set. However, as mentioned above, this set does not even map to an abstract complete set, therefore this abstraction cannot work.

A careful reader might wonder: do we have to include the set $\{a, b, \hat{c}\}$ in the abstract complete sets? Could it be the case that this set is always spurious? As an answer to this, we present the following framework:

$$\mathcal{L} = \{\neg c, \neg b\} \cup \mathcal{A},\ \mathcal{A} = \{a, b, c, c'\},$$
$$\mathcal{R} = \{\neg c \leftarrow a, \neg b \leftarrow c\},\ \overline{b} = \neg b, \overline{c} = \neg c.$$

This framework maps to the same $\hat{D}$ under the same map m, and in this framework the set $\{a, b, c'\}$ is complete. This justifies accepting $\{a, b, \hat{c}\}$ as an abstract complete set.

As in classical ABAFs, an atom is credulously accepted, under $\hat{\sigma}$, if there is a $\hat{\sigma}$ set deriving the queried atom.

Proposition 6. *Let $\hat{D}$ w.r.t. $\hat{\sigma}$ be faithful to D w.r.t. some semantics σ, and $\hat{a} \in \hat{\mathcal{A}}$.*

1. *If $\hat{a}$ is credulously accepted in $\hat{D}$, then there is an assumption $a \in \hat{a}$ so that a is credulously accepted in D.*
2. *If $\hat{a}$ is not in any $\hat{A} \in \sigma(\hat{D})$, then for every $a \in \hat{a}$, it holds that a is not in any $A \in \sigma(D)$.*

If there is an $\hat{A} \in \hat{\sigma}(\hat{D})$ s.t. $Single(\hat{A}) = \hat{A}$, then $x \in Th_{\hat{D}}(\hat{A})$ iff x is credulously accepted in D under σ.

In the case that involves singletons, the above says that given a faithful clustering that maps an assumption a to itself, the fact that a is credulously accepted in the cABAF, can be translated into the fact that a is credulously accepted in the classical framework.

An atom x that is credulously accepted under clustered semantics in $\hat{D}$, is not necessarily credulously derived in D as well. Proposition 1 implies that for a given clustered set $\hat{A}$ that derives an atom x, there is a classical set A that maps to $\hat{A}$ and also derives x. However, if $\hat{A} \in \hat{\sigma}(\hat{D})$ for some semantics σ, that means that x is credulously accepted in $\hat{D}$, but there is no guarantee $A \in \sigma(D)$, hence x might not be credulously derivable in D. By requiring credulous acceptance to stem from sets of singletons only, implies, by Proposition 1, that acceptance transfers to the concrete case.

5 Obtaining Clusterings and Use Cases

In this section we present two methods for obtaining faithful clusterings and example uses. The first method starts at a "coarse" abstraction, e.g., with clustering all assumptions into one big cluster and refines, upon user requests using answer set programming (ASP) (Brewka, Eiter, and Truszczyński 2011). The other approach computes a faithful clustering under adm starting with singletons and iteratively clusters following the grounded semantics.

Abstraction & refinement We implemented a prototype tool, which is a modification of the tool provided for clustering arguments in AFs (Saribatur and Wallner 2021) (which, in turn, is based on work on abstraction in ASP (Saribatur, Eiter, and Schüller 2021)). This tool conducts an abstraction & refinement algorithm to find a faithful abstraction in a given cABAF. That is, this tool follows the prominent counterexample guided abstraction refinement (CEGAR) approach (Clarke et al. 2003). The candidate space (possible clusterings) is allowing also clusterings that are spurious. By checking spuriousness, we can either conclude that the given clustering is faithful, which can be returned, or is not faithful. A counterexample to faithfulness is a reason for spuriousness that can be used to for finding hints how to refine the search space of clusterings. That is, how to generate the next clustering. This process is repeated until a faithful clustering is found.

The tool is interactive: if a faithful abstraction is obtained, the tool prompts the user with the found mapping and asks whether they would need further details on some assumptions in a cluster. If yes, the tool continues the search for a faithful abstraction from the input the user provides.

Checking faithfulness is not trivial, and we implemented this check using ASP encodings of the abstract argumentation semantics. These encodings are based on previous ASP encodings that compute assumption sets on classical ABAFs (Lehtonen, Wallner, and Järvisalo 2021). We disabled further abstractions of the original tool.

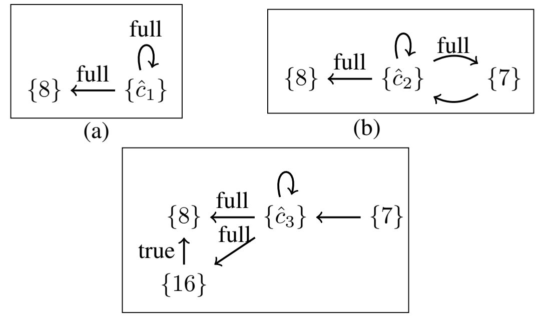


Figure 4: Attacks among sets containing one (clustered) assumption in *aba_100_0.3_5_5_1.aba*. The first clustering (a) $\hat{\mathcal{A}}_1 = \{8, \hat{c}_1\}$ clusters all assumptions but one. Set $\{8, \hat{c}_1\}$ is abstract admissible. The second clustering (b) is $\hat{\mathcal{A}}_2 = \{7, 8, \hat{c}_2\}$ and assumption set $\{7, 8\}$ is (spuriously) abstract admissible. In (c) we have $\hat{\mathcal{A}}_3 = \{7, 8, 16, \hat{c}_3\}$ and $\{7, 8\}$ ceases to be abstract admissible: the true attack from $\{16\}$ is not counter-attacked by $\{7, 8\}$.

Uses Cases Let us look at two ABAFs from the recent 5th International Competition on Computational Models of Argumentation (ICCMA)[2] without any stable assumption set: *aba_100_0.1_10_10_5.aba* and *aba_100_0.3_5_5_1.aba*. The first one contains ten assumptions and the second one thirty assumptions.

We first look at reasons why there are no stable assumptions in the smaller instance. As computed by our prototype, refining the clustering with all assumptions in one big cluster, we arrive at a faithful clustering that only makes one assumption concrete: 3. Inspecting this assumption one sees that $\overline{3} = 3$, and in flat ABA frameworks assumptions (like 3) cannot be derived. That is, there can be no stable assumption in this ABAF.

For the larger instance, our prototype tool suggested an initial clustering, which we manually refined further. The resulting faithful clustering makes six assumptions concrete (3, 6, 13, 16, 20, and 22). Inspecting this instance, it becomes clear that 22 is self-attacking, but the contrary $\overline{22}$ can be derived. However, only conflicting sets, in the clustering, attack 22, directly indicating that no stable assumption sets can exist here. Note that the abstraction reduces the number of assumptions to see this behavior from thirty to six.

Next, let us look at the larger instance and the credulously accepted assumption 8 under admissibility. Figure 4 shows

[2]https://iccma2023.github.io/

attacks, full attacks, and true attacks among assumption sets containing one (clustered) assumption.[3] Clustering all assumptions except 8 into one big cluster $\hat{c}_1$ (Figure 4(a)) is deemed faithful by our tool (under $\hat{adm}$). Indeed, we get $\{8, \hat{c}_1\}$ as an abstract admissible set, mirroring classical admissible sets on the original framework.

Let us "zoom in" more. Since 7 contributes to deriving $\overline{8}$, we proceed by making 7 concrete (singleton), as well. That is, we have now 7, 8, and $\hat{c}_2$ as the (clustered) assumptions (Figure 4(b)). This is deemed not faithful: our abstract admissible semantics states that $\{7, 8\}$ is abstract admissible. However, this set is not admissible in the original framework; when looking at attacks among sets containing one (clustered) assumption: $\{7\}$ defends $\{8\}$ against $\{\hat{c}_2\}$.

The refinement tool in addition suggests making 16 concrete to arrive at a faithful clustering (Figure 4(c)). On this clustering the (faithful) abstract admissible sets imply that 7 and 8 only occur together if $\hat{c}_3$ is also present. As shown in the figure, 16 truly attacks 8, and $\{\hat{c}_3\}$ defends 8 against 16. The two assumptions 7 and 8 do not fully attack anything and do not attack 16 in this clustering, so they need to be defended by $\hat{c}_3$. This clustering tells us that 7 and 8 can, in fact, be accepted jointly, but require additional defense from "inside" $\hat{c}_3$ that we abstracted away.

Iterative clustering Next, we present a way of clustering that is faithful under adm. The idea is to cluster w.r.t. defended sets, mimicking the grounded assumptions set in ABAFs (Čyras et al. 2018). Both the grounded assumption set and the clustering can be computed in polynomial time.

We create three types of clusters, the d- , a- and s-type. In d_0 we include all unattacked assumptions, and in d_i all assumptions defended by all previous $d-$type clusters. In other words, we follow the following scheme:

$$d_0 = m(\mathit{DefendedBy}(\emptyset)),$$
$$d_i = m(\mathit{DefendedBy}(\bigcup_{j=0}^{i-1} d_j)) \setminus \bigcup_{j=0}^{i-1} d_j.$$

Clusters of type a are singletons that are attacked by some $d-$ type cluster. We denote by a_{i_j} a singleton attacked by d_i, for some $j \in \mathbb{N}$. All those assumptions that are not $d-$ or $a-$ type, are mapped to singletons $s_1, \ldots, s_l$. We call this specific clustering, defence-based clustering.

Proposition 7. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, and m be the defence-based clustering. m is faithful under admissibility.*

Example 7. *Let us look again to Example 2. First, we find the assumptions that are not attacked. In this case, these assumptions are e and b, hence the first cluster is $\hat{d}_0 = \{e, b\}$. Then we check which assumptions do they defend. Since e attacks d, and d is the only way to derive the contrary of a, i.e. z, then the set $\{e, b\}$ defends a. This is the only defended assumption, thus $\hat{d}_1 = \{a\}$. We also observe that assumption d is attacked by cluster $\hat{d}_0$, and assumption c is attacked by both $\hat{d}_0$ and $\hat{d}_1$, therefore they are $a-$ type clusters, namely a_0 and a_1, respectively. As a matter of fact, this clustering is indeed faithful, since the admissible sets $\{a, b, e\}, \{a, e\}, \{b, e\}, \{e\}, \{b\}, \emptyset$ are mapped faithfully to the sets $\{\hat{d}_0\}, \{\hat{d}_0, \hat{d}_1\}$, and $\emptyset$.*

[3]Attacks involving assumption sets with more members are not shown due to conciseness of presentation.

6 Relating ABAFs and AFs

In this section we relate to the recent clustering (Saribatur and Wallner 2021) on AFs (Dung 1995).

An AF $F = (\mathit{Args}, \mathit{Att})$ is a pair of a set of arguments Args and $\mathit{Att} \subseteq \mathit{Args} \times \mathit{Args}$ represents attacks. We say that an argument $a \in \mathit{Args}$ is defended by a set $S \subseteq \mathit{Args}$, iff $\forall b \in \mathit{Args}$ s.t. $(b, a) \in \mathit{Att}, \exists c \in S$ s.t. $(c, b) \in \mathit{Att}$. This definition extends naturally to defending sets of arguments. A set $S \subseteq \mathit{Args}$ is conflict-free, iff $\forall a, b \in S, (a, b) \notin \mathit{Att}$, and admissible, iff $S \in \mathit{cf}(F)$ and S defends itself.

Given a surjective mapping m, we define the clustered AF $\hat{F}$ to be the pair $(\hat{\mathit{Args}}, \hat{\mathit{Att}})$, where $m(\mathit{Args}) = \hat{\mathit{Args}}$ and $m(\mathit{Att}) = \hat{\mathit{Att}} = \{(\hat{a}, \hat{b}) : (a, b) \in \mathit{Att}, m(a) = \hat{a}, m(b) = \hat{b}\}$. A clustered argument a is called a singleton iff $|\hat{a}| = 1$.

Let $\hat{F} = (\hat{\mathit{Args}}, \hat{\mathit{Att}})$ be a clustered AF according to m. The set $\hat{S} \subseteq \hat{\mathit{Args}}$ is conflict-free, iff $\forall \hat{a}, \hat{b} \in \mathit{Single}(\hat{S})$ we have $(\hat{a}, \hat{b}) \notin \hat{\mathit{Att}}$, admissible, iff $\hat{S} \in \hat{\mathit{cf}}(\hat{F})$ and the set $\mathit{Single}(\hat{S})$ is defended.

In the case of AFs, the clustering happens on the level of arguments. As a first comparison, abstracting on structured frameworks provokes slightly more involved notions than in non-structured frameworks. For example, attacks on clustered AFs require knowledge of singletons, while in ABAFs we have three types of attacks.

Interestingly, structure allows us to take advantage of the number of vulnerabilities of a cluster. In a clustered AF, consider a cluster that attacks another cluster. We only know that there is an argument in the first cluster that attacks some other argument in the second cluster. However, it could be the case that there are more arguments in the first cluster, that address attacks in one or more arguments in the second cluster. In ABAFs, structure offers us the benefit to handle the relations of derivability in a more sophisticated way, and thus gain more information on attacks. Even though when clustering assumptions we lose the details on which assumption is "responsible" for deriving an atom, we still have the information that the atom can be derived by a specific set.

Example 8. *We consider the frameworks of our running Example 2. Viewed as an AF, this AF F contains four arguments, A, C, D, and E, representing the rules $a, b \leftarrow x$, $c \leftarrow y$, $d \leftarrow z$, and $e \leftarrow w$, respectively. The attacks in F are given by the set $\mathit{Att} = \{(E, D), (D, A), (A, C)\}$. Then we apply the mapping m as follows, $m(A) = A, m(C) = \hat{C}, m(D) = \hat{C}, m(E) = E$. The set $\{A, E, \hat{C}\}$ is conflict-free in $\hat{F}$ w.r.t. the abstract AF semantics, as the attacks in $\hat{F}$, i.e., $(A, \hat{C}), (\hat{C}, A), (E, \hat{C})$, are not attacks among singletons. However, the sets $\{A, C, D, E\}$, $\{A, D, E\}$, and $\{A, C, E\}$ (preimages of the set above) are not conflict-free. Hence this set is spurious.*

Let us take into account the information that lies in the structure of the ABAF. The sets $\{a, b, c, d, e\}$, $\{a, b, c, e\}$, and $\{a, b, d, e\}$ are not conflict-free in D. Clustering c, d to cluster $\hat{c}$, we get that the set $\{a, b, \hat{c}, e\}$ is conflicting under $\hat{cf}$. Hence this set is not spurious, indicating that structure can be beneficial to minimize spuriousness. Note that $\{a, b, \hat{c}, e\}$ corresponds to $\{A, E, \hat{C}\}$ on the AF-level.

Moreover, in contrast to the AF case optimality of abstract admissibility is more intricate: we showed NP-hardness when one would design an optimal abstract admissible semantics (Section 4). Finally, as shown previously (Lehtonen et al. 2023), AF and ABA reasoning also differs in terms of the number of arguments generated. When considering the direct instantiation of AFs from ABAFs, the number of arguments is in general not polynomially bounded by the given ABAF. Abstraction on ABAFs then can operate on potentially smaller structures to begin with.

7 Conclusions

In this work we introduced existential abstraction to assumption-based argumentation (ABA), by clustering assumptions. Such clusterings allow for abstracting parts of the given knowledge. We proposed semantics of the resulting clusterings. We showed that our definition of conflict-freeness is optimal in avoiding spuriousness. For admissibility, we showed that such optimal criteria are NP-hard to check. For our semantics, we presented use cases and two approaches to obtain (faithful) clusterings: one via using a "grounded-like" algorithm and one using an iterative refinement, which can be queried interactively by a user.

We believe that our approach can be beneficial for supporting explainability, a key area of formal argumentation, by providing foundational work towards abstracting certain parts of argumentative reasoning in a faithful manner. Interactive tools that give users the ability to "zoom in" or "zoom out" can be useful to improve understanding. Among interesting avenues for future works are, e.g., extending our approach to other formal approaches to structured argumentation, such as ASPIC$^+$ (Modgil and Prakken 2018), defeasible logic programming (DeLP) (García and Simari 2018), deductive argumentation (Besnard and Hunter 2018), Carneades (Gordon, Prakken, and Walton 2007), or Gorgias (Kakas, Moraitis, and Spanoudakis 2019). Moreover, extending abstraction with a recently proposed notion of forgetting parts of an ABA knowledge base (Berthold, Rapberger, and Ulbricht 2023) appears intriguing. Finally, existential abstraction via clustering was studied also for ASP (Saribatur, Eiter, and Schüller 2021) via modification of logic programs and subsequent use of answer set semantics. We believe applying the approach followed in this paper, i.e., only minimally modifying the given structure (logic program) and usage of an "abstract answer set semantics" can be lead to an interesting future research direction for abstraction in ASP.

Acknowledgements

This research was funded in whole or in part by the Austrian Science Fund (FWF) by grants P35632 and T-1315.

References

Amgoud, L.; Dimopoulos, Y.; and Moraitis, P. 2007. A unified and general framework for argumentation-based negotiation. In Durfee, E. H.; Yokoo, M.; Huhns, M. N.; and Shehory, O., eds., *Proc. AAMAS*, 967–974. IFAAMAS.

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G. R.; Thimm, M.; and Villata, S. 2017. Towards artificial argumentation. *AI Mag.* 38(3):25–36.

Baroni, P.; Boella, G.; Cerutti, F.; Giacomin, M.; van der Torre, L. W. N.; and Villata, S. 2014. On the input/output behavior of argumentation frameworks. *Artif. Intell.* 217:144–197.

Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds. 2018. *Handbook of Formal Argumentation*. College Publications.

Baroni, P.; Caminada, M.; and Giacomin, M. 2011. An introduction to argumentation semantics. *Knowl. Eng. Rev.* 26(4):365–410.

Bench-Capon, T. J. M. 2016. Dilemmas and paradoxes: cycles in argumentation frameworks. *J. Log. Comput.* 26(4):1055–1064.

Berthold, M.; Rapberger, A.; and Ulbricht, M. 2023. Forgetting aspects in assumption-based argumentation. In Marquis, P.; Son, T. C.; and Kern-Isberner, G., eds., *Proc. KR*, 86–96. ijcai.org.

Besnard, P., and Hunter, A. 2018. A review of argumentation based on deductive arguments. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 9, 437–484.

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.; Eiter, T.; and Truszczyński, M. 2011. Answer set programming at a glance. *Commun. ACM* 54(12):92–103.

Clarke, E.; Grumberg, O.; Jha, S.; Lu, Y.; and Veith, H. 2003. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM* 50(5):752–794.

Čyras, K.; Fan, X.; Schulz, C.; and Toni, F. 2018. Assumption-based argumentation: Disputes, explanations, preferences. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 7, 365–408.

Cyras, K.; Oliveira, T.; Karamlou, A.; and Toni, F. 2021a. Assumption-based argumentation with preferences and goals for patient-centric reasoning with interacting clinical guidelines. *Argument Comput.* 12(2):149–189.

Cyras, K.; Rago, A.; Albini, E.; Baroni, P.; and Toni, F. 2021b. Argumentative XAI: A survey. In Zhou, Z., ed., *Proc. IJCAI*, 4392–4399. ijcai.org.

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., *Proc. AAMAS*, 1252–1260. IFAAMAS.

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.; Järvisalo, M.; Linsbichler, T.; Niskanen, A.; and Woltran, S. 2019. Preprocessing argumentation frameworks via replacement patterns. In Calimeri, F.; Leone, N.; and Manna, M., eds., *Proc. JELIA*, volume 11468 of *Lecture Notes in Computer Science*, 116–132. Springer.

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., *Proc. ECAI*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, 324–329. IOS Press.

Fan, X., and Toni, F. 2015. On computing explanations in argumentation. In Bonet, B., and Koenig, S., eds., *Proc. AAAI*, 1496–1502. AAAI Press.

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., *Proc. AAMAS*, 1153–1161. ACM.

García, A. J., and Simari, G. R. 2018. Argumentation based on logic programming. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 8, 409–435.

Gordon, T. F.; Prakken, H.; and Walton, D. 2007. The Carneades model of argument and burden of proof. *Artif. Intell.* 171(10-15):875–896.

Kakas, A. C.; Moraitis, P.; and Spanoudakis, N. I. 2019. *GORGIAS*: Applying argumentation. *Argument Comput.* 10(1):55–81.

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., *Proc. KR*, 440–450.

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

Modgil, S., and Prakken, H. 2018. Abstract rule-based argumentation. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 6, 287–364.

Prakken, H., and Sartor, G. 2015. Law and logic: A review from an argumentation perspective. *Artif. Intell.* 227:214–245.

Prakken, H., and Winter, M. D. 2018. Abstraction in argumentation: Necessary but dangerous. In Modgil, S.; Budzynska, K.; and Lawrence, J., eds., *Proc. COMMA*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, 85–96. IOS Press.

Prakken, H. 2023. Relating abstract and structured accounts of argumentation dynamics: the case of expansions. In Marquis, P.; Son, T. C.; and Kern-Isberner, G., eds., *Proc. KR*, 562–571. ijcai.org.

Rapberger, A., and Ulbricht, M. 2023. On dynamics in structured argumentation formalisms. *J. Artif. Intell. Res.* 77:563–643.

Sakama, C. 2018. Abduction in argumentation frameworks. *J. Appl. Non Class. Logics* 28(2-3):218–239.

Saribatur, Z. G., and Wallner, J. P. 2021. Existential abstraction on argumentation frameworks via clustering. In Bienvenu, M.; Lakemeyer, G.; and Erdem, E., eds., *Proc. KR*, 549–559. ijcai.org.

Saribatur, Z. G.; Eiter, T.; and Schüller, P. 2021. Abstraction for non-ground answer set programs. *Artif. Intell.* 300:103563.

Toni, F. 2014. A tutorial on assumption-based argumentation. *Argument Comput.* 5(1):89–117.

Ulbricht, M., and Wallner, J. P. 2022. Strongly accepting subframeworks: Connecting abstract and structured argumentation. In Toni, F.; Polberg, S.; Booth, R.; Caminada, M.; and Kido, H., eds., *Proc. COMMA*, volume 353 of *Frontiers in Artificial Intelligence and Applications*, 320–331. IOS Press.

Vassiliades, A.; Bassiliades, N.; and Patkos, T. 2021. Argumentation and explainable artificial intelligence: a survey. *Knowl. Eng. Rev.* 36:e5.

Wallner, J. P. 2020. Structural constraints for dynamic operators in abstract argumentation. *Argument Comput.* 11(1-2):151–190.