

Existential Abstraction on Argumentation Frameworks via Clustering

Zeynep G. Saribatur¹, Johannes P. Wallner²

¹Institute of Logic and Computation, TU Wien, Austria

²Institute of Software Technology, Graz University of Technology, Austria
zeynep@kr.tuwien.ac.at, wallner@ist.tugraz.at

Abstract

Argumentation in Artificial Intelligence (AI) builds on formal approaches to reasoning argumentatively. Common to many such approaches is to use argumentation frameworks (AFs) as reasoning engines, with AFs being composed of arguments and attacks between arguments, which are instantiated from knowledge bases in a principle-based manner. While representing what can be argued for in an AF provides a conceptually clean way, this process can face challenges arising from generating a large number of arguments, which can act as a barrier to explainability. Inspired by successful approaches to model checking where the state explosion is mitigated by applying existential abstraction, we study an adaption of existential abstraction in form of clustering arguments in an AF to address an associated "argument explosion". In this paper, we provide a foundational investigation of this form of existential abstraction by defining semantics of the resulting clustered AFs, which balance two inherent aspects of existential abstractions: abstracting from concrete AFs and not permitting too much spuriousness (i.e., conclusions that hold on the abstraction but not on the original AF). Moreover, we show properties of clustered AFs, including complexity results, discuss use of clusterings for explaining results of reasoning tasks, and employ the recently introduced methodology of abstraction in answer set programming (ASP) for obtaining and reasoning over clustered AFs.

1 Introduction

Computational models of argumentation (Baroni et al. 2018) provide versatile formal approaches to non-monotonic and argumentative reasoning within Artificial Intelligence (AI), and find heterogeneous application venues such as in legal and medical reasoning (Atkinson et al. 2017). In order to provide rational accounts of what can be argued for, approaches to what is called structured argumentation prescribe formal workflows to argumentative reasoning. From a knowledge base ways of generating arguments and their relationships are defined. Based on these acceptable sets of arguments can be identified. The most prominent approach to model arguments and their relations is to use argumentation frameworks (AFs) (Dung 1995), which consist of arguments and a binary attack relation between arguments, interpreted as a counter-argument relation. Argumentation semantics specify criteria to find sets of arguments in an AF that can be deemed acceptable, from an argumentative standpoint.

A prominent criterion for argumentation semantics is that of admissibility. A set of arguments in an AF is called admissible if this set is conflict-free (no attacks between arguments in the set), and each attack onto this set is countered from inside the set. As an example for an important reasoning task, one can ask whether an argument is credulously accepted under admissibility, i.e., whether there is an admissible set containing this argument (implying that one can coherently argue in favour of this argument).

While formal workflows to obtain AFs allow for satisfaction of rationality postulates in subsequent argumentative reasoning (Caminada 2018), current approaches to AF construction face the inherent challenge of generating large AFs. For instance, the number of arguments in an AF might not be polynomially bounded by (the size of) a knowledge base the AF was constructed from, even when considering optimizations or restrictions (Yun, Vesic, and Croitoru 2018; Amgoud, Besnard, and Vesic 2014; Lehtonen, Wallner, and Järvisalo 2017). Recent competitions (Thimm and Villata 2017; Gaggl et al. 2020; Bistarelli et al. 2020) show that, from a computational perspective, scalability is increasing, and systems are capable of solving argumentative reasoning tasks in an automated manner for larger and larger AFs. Nevertheless, large AFs still pose a barrier to applicability of computational models of argumentation. It remains difficult to digest large AFs, even when, e.g., admissible sets can be found efficiently: both human and artificial agents working on AFs face the barrier of having to handle many arguments or large admissible sets (for artificial agents this can be an issue, e.g., when using conclusions in processes later on).

1.1 Motivation for Existential Abstraction

In similar terms to what is referred to as "state explosion" in complex systems, we face the issue of "argument explosion". In this paper, we use this opportunity to introduce and study a concept that has been successfully employed to address the state explosion problem: existential abstraction, prominently used in model checking (Clarke, Grumberg, and Long 1994; Clarke et al. 2003). Before continuing we need a precursor to our discussion: the term "abstraction" is already in use when referring to AFs, usually meaning that the arguments in an AF are seen as "abstract entities", i.e., they are seen as entities without further (internal) structure. In contrast, in this paper we mean a differ-

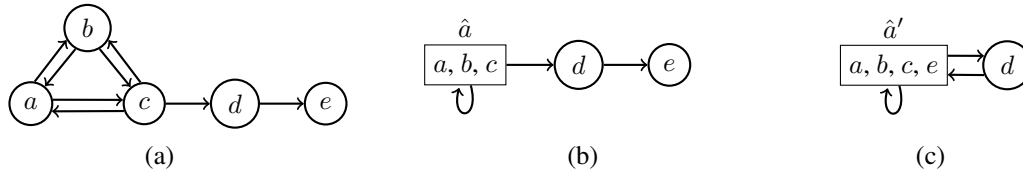


Figure 1: An example AF (a), a faithful clustered AF (b), and a clustered AF with a spurious admissible set (c).

ent concept of abstraction. We adapt existential abstraction found in model checking to AFs by studying a formally defined approach that abstracts AFs, resulting in “abstracted” AFs. Concretely, we apply existential abstraction by *clustering arguments* in an AF. We assume finite sets of arguments.

A convenient way of clustering of arguments in an AF is to consider a (surjective) mapping $m : A \rightarrow \hat{A}$, which maps arguments A in an AF to a set \hat{A} . We exemplify such a clustering (formal definitions follow in the main part).

Example 1. Consider an AF composed of five arguments a, b, c, d , and e , and attacks as shown in Figure 1(a). Say, we map arguments a, b , and c to a cluster \hat{a} , and both d and e to themselves. The resulting “clustered AF” can be seen in Figure 1(b). The attacks involving clustered arguments are interpreted existentially: if clustered argument \hat{a} attacks an argument d , then in the AF mapped to the clustered AF there exists an argument inside the cluster attacking b . In our case c , which is mapped to \hat{a} , attacks d .

Naturally, abstractions are useful if certain properties are preserved from the structures we abstract from. In order to study properties, or semantics, that can be preserved, we need to answer a key question: how to interpret clustered AFs, from a semantical viewpoint?

Starting from an intuitive view, what is the concept of an admissible set within a clustered AF? We can use the existential flavour of the abstraction to answer this question. We abstracted away knowledge about the exact attack structure involving clustered arguments. That is, we need to be prepared to assume *some* attack structure *compatible* with the clustered AF. Considering again Example 1 (Figure 1(b)), we can infer that there exists an attack from a, b , or c onto d , and at least one attack among a, b , and c in a concrete AF mapped to the clustered AF, but which one we cannot directly infer from the clustered AF due to the abstraction. Thus, our notion of existential abstraction involves loss of information (common to abstraction). Reasoning further, there can be an argument in the cluster that defends itself (or is unattacked) and attacks c , thereby defending d . Faced with the information stored in the clustered AF we are required to list $\{\hat{a}, d\}$ as a “potential admissible set” (more precisely, one argument in the cluster and d). This contrasts classical admissibility: \hat{a} would be considered self-attacking in a classical AF. In the AF in Figure 1(a), a together with d is an admissible set. That is, $\{\hat{a}, d\}$ in fact corresponds to an admissible set in the “original” AF.

Equipped with a notion of “potential admissible sets” on clustered AFs, we can specify existential abstraction in way such that if an AF F is mapped to a clustered AF \hat{F} , then

each admissible set in F can be mapped to a “potential admissible set” in \hat{F} (preserving admissible sets).

However, while abstractions preserve interesting features, there is a constant companion of abstractions: spuriousness.

Example 2. Consider a different mapping for the AF in Figure 1(a): m' maps all of a, b, c , and e to a cluster \hat{a}' and d to itself. The corresponding clustered AF can be seen in Figure 1(c). If we reason similar as before, then we need to accept $\{d\}$ as a potential admissible set (e.g., d defends itself against all attacks).

Viewing $\{d\}$ as a potential admissible set is misleading, since in the original AF there is no admissible set $\{d\}$. We say that such a set is *spurious* w.r.t. the AF from Figure 1(a). In contrast, under the formal definitions to follow, the clustered AF in Figure 1(b) does not have any spurious admissible set w.r.t. the AF in Figure 1(a). That is, the clustered AF is *faithful* w.r.t. this AF.

On faithful clustered AFs we can see an interesting aspect: faithfulness implies that there is an admissible set containing e in the original AF mapped to the clustered AF. We can infer that one argument in the cluster \hat{a} defends e against d , but the concrete information which one is responsible for defense is abstracted away. In this way, we arrive at a simplified AF (the clustered AF) that shows properties of concrete AFs, but without providing all details, thereby leading to an approach and methodological tool that can be used to obtain simplifications of AFs. For instance, we envision that (existential) abstractions can be used for presenting results of automated argumentative reasoning to (human) users. For instance, visualization methods of formalized arguments could benefit from such approaches.

1.2 Contributions

Our main contribution is a formal study of clustering AFs:

- we formally define clustering on AFs with the notions of being an abstraction, being spurious, and being faithful;
- we define conflict-free sets, admissible sets, and stable extensions on clustered AFs that are *optimal* in the sense of being abstracting, by permitting as few sets (extensions) as possible to reduce potential spuriousness;
- we show properties of the (abstract) semantics of clustered AFs, including complexity results;
- we discuss the use of clustering for explaining results of reasoning tasks on AFs;
- by employing the recently introduced methodology on ASP abstraction, we construct such explanatory clustered AFs that can be reasoned over using the novel semantics.

2 Argumentation Frameworks

Argumentation frameworks (AFs) were introduced by Dung (1995), and represent arguments as vertices and attacks between arguments as directed edges in a graph.

Definition 1. An argumentation framework (AF) is a pair $F = (A, R)$, where A is a finite set of arguments and $R \subseteq A \times A$ is the attack relation. The pair $(a, b) \in R$ means that a attacks b .

We sometimes refer to AFs as “classical” AFs, to distinguish them from the clustered AFs we introduce in the next section. A set $S \subseteq A$ attacks an $a \in A$ (in F) if there is a $(b, a) \in R$ with $b \in S$.

Definition 2. Let $F = (A, R)$ be an AF. An argument $a \in A$ is defended (in F) by a set $S \subseteq A$ if for each $b \in A$ such that $(b, a) \in R$ there exists $c \in S$ such that $(c, b) \in R$.

Semantics (Baroni, Caminada, and Giacomin 2011) for argumentation frameworks are defined through a function σ which assigns to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions. We consider for σ the functions *cf*, *adm*, *com*, *grd*, *stb*, and *prf*, which stand for conflict-free, admissible, complete, grounded, stable, and preferred, respectively. Towards the definition we make use of the characteristic function of AFs, defined for an AF $F = (A, R)$ by $\mathcal{F}_F(S) = \{x \in A \mid x \text{ is defended by } S\}$.

Definition 3. Let $F = (A, R)$ be an AF. An $S \subseteq A$ is conflict-free (in F) if there are no $a, b \in S$ such that $(a, b) \in R$. We denote the set of conflict-free sets of F by $cf(F)$. For an $S \in cf(F)$ it holds that

- $S \in adm(F)$ iff $S \subseteq \mathcal{F}_F(S)$;
- $S \in com(F)$ iff $S = \mathcal{F}_F(S)$;
- $S \in grd(F)$ iff S is the least fixed-point of \mathcal{F}_F ;
- $S \in stb(F)$ iff S attacks in F each $a \in A \setminus S$; and
- $S \in prf(F)$ iff $S \in adm(F)$ and $\nexists T \in adm(F)$ s.t. $S \subset T$.

For an AF $F = (A, R)$, an argument $a \in A$ is said to be credulously accepted under σ if there is an $E \in \sigma(F)$ with $a \in E$. In contrast, a is skeptically accepted under σ if $a \in E$ holds for all $E \in \sigma(F)$.

3 Clustered Argumentation Frameworks

We formally introduce existential abstraction by first presenting clustered argumentation frameworks, the outcome of clustering. Main component of clustering is a mapping on arguments.

Definition 4. Given a set A of arguments, an argument mapping m is a surjective mapping $m : A \mapsto \hat{A}$ for a set \hat{A} of clustered arguments.

Surjectivity implies $|\hat{A}| \leq |A|$. For convenience, we extend the mapping to sets of arguments and set of sets of arguments (like semantics of AFs), as follows. For a set $S \subseteq A$ we define $m(S) = \{m(a) \mid a \in S\}$. For a set of sets $\mathcal{E} \subseteq 2^A$ we define $m(\mathcal{E}) = \{m(S) \mid S \in \mathcal{E}\}$.

A clustered AF is formally speaking an AF according to a mapping m , which specifies arguments that are mapped onto the clusters.

Definition 5. An AF $\hat{F} = (\hat{A}, \hat{R})$ is a clustered AF according to a mapping m if \hat{A} is the image of m .

Mapping a given (classical) AF using a mapping m is then direct, defined formally next.

Definition 6. Let $F = (A, R)$ be an AF, and m a mapping on A . The (unique) clustered AF $m(F) = \hat{F} = (\hat{A}, \hat{R})$ according to m is given by $m(A) = \hat{A}$ and $m(R) = \hat{R} = \{(\hat{a}, \hat{b}) \mid (a, b) \in R, m(a) = \hat{a}, m(b) = \hat{b}\}$.

That is, arguments in an AF $F = (A, R)$ are mapped, according to a mapping m on A , into clusters, which constitute the arguments of the associated clustered AF. An attack (a, b) in R is then mapped into $(m(a), m(b))$ (like arguments, multiple attacks might map to the same attack). We call arguments in a clustered AF also clustered arguments.

Whenever convenient, and when the mapping m is clear from the context, we identify a clustered argument $\hat{a} \in \hat{A}$ with the set $m^{-1}(\hat{a}) = \{a \mid m(a) = \hat{a}\}$, i.e., the set of arguments mapped onto \hat{a} . A clustered argument $\hat{a} \in \hat{A}$ is called a *non-singleton cluster* if there exists more than one “original” argument $a_1, \dots, a_n \in A, n > 1$, s.t. $m(a_i) = \hat{a}, 1 \leq i \leq n$ (equivalently when using sets, $|\hat{a}| > 1$). Otherwise, \hat{a} is a *singleton*, i.e., $|\hat{a}| = 1$. For a clustered AF $\hat{F} = (\hat{A}, \hat{R})$ we define the shorthand $single_m(\hat{A}) = \{\hat{a} \in \hat{A} \mid |\hat{a}| = 1\}$, which contains all singleton clustered arguments, and omit m if clear from the context. For convenience, from now on, we assume that singletons are always mapped to themselves, i.e., if $x \in single(\hat{A})$ then $x \in A$ and $m(x) = x$.

Intuitively, the pair $(\hat{a}, \hat{b}) \in \hat{R}$ means that there exists some $a \in m^{-1}(\hat{a})$ that attacks some $b \in m^{-1}(\hat{b})$, reflecting the underlying existential abstraction. There can be multiple AFs mapped to a single clustered AF. In particular, there can also be multiple AFs mapped onto a single clustered AF even under a single mapping m (the attack relation of the original AFs can be different).

Example 3. Consider again the AF $F = (A, R)$ from Figure 1(a) which is formally given by $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (b, a), (b, c), (c, b), (a, c), (c, a), (c, d), (d, e)\}$. Define mapping $m : A \rightarrow \hat{A}$ with $\hat{A} = \{\hat{a}, d, e\}$ and $m(a) = m(b) = m(c) = \hat{a}, m(d) = d, \text{ and } m(e) = e$. The clustered AF $m(F) = \hat{F}$ according to m is the one in Figure 1(b). We have $m^{-1}(\hat{a}) = \{a, b, c\}$. If one would remove one attack among a, b , and c in F , resulting in F' , then $m(F') = \hat{F}$ remains the same (there still exists an attack from one argument in $\{a, b, c\}$ to one of these arguments). That is, both F and F' are mapped, under the same m , to the same clustered AF \hat{F} . Considering $m'(a) = m'(b) = m'(c) = m'(e) = \hat{a}$ and $m'(d) = d$ one obtains $m'(F) = \hat{F}'$, from Figure 1(c).

4 Semantics for Clustered AFs

Syntactically, clustered AFs are close to classical AFs. Nevertheless, classical semantics of AFs are not applicable to clustered AFs within the existential abstraction methodology. As an intuitive example, if a clustered argument is not a singleton, then certain information about the arguments

clustered in this clustered argument is abstracted away, making standard AF semantics inapplicable. Towards semantics of clustered AFs, we first define such semantics in a general manner (in order to study such semantics formally).

Definition 7. A semantics $\hat{\sigma}$ on clustered AFs returns a set of sets of arguments $\hat{\sigma}(\hat{F}, m) \subseteq 2^{\hat{A}}$, for any given clustered AF $\hat{F} = (\hat{A}, \hat{R})$ according to a mapping m .

Again, if the mapping m is clear from the context, we simply write $\hat{\sigma}(\hat{F})$. However, we remark that it will be important for a semantics to distinguish, in particular, between clustered arguments that are singletons and those that are not. Simply put, attacks between singletons imply that no information (involvement of attacks regarding this argument) was abstracted away.

Next, we define central properties of clustered AFs, reflecting central notions of existential abstractions.

Definition 8. Let $F = (A, R)$ be an AF, m a mapping on A , $m(F) = \hat{F}$, σ a semantics on AFs, and $\hat{\sigma}$ a semantics on clustered AFs. According to m , we say that

- \hat{F} under $\hat{\sigma}$ abstracts F under σ if $m(\sigma(F)) \subseteq \hat{\sigma}(\hat{F})$,
- $\hat{E} \in \hat{\sigma}(\hat{F})$ is spurious w.r.t. F under σ if $\nexists E \in \sigma(F)$ s.t. $m(E) = \hat{E}$, and
- \hat{F} under $\hat{\sigma}$ is faithful w.r.t. F under σ if there is no spurious $\hat{E} \in \hat{\sigma}(\hat{F})$ w.r.t. F under σ .

In words, a clustered AF \hat{F} under a clustered semantics $\hat{\sigma}$ abstracts a classical AF F under a classical semantics σ if for each σ -extension E of F it holds that mapping this σ -extension under m leads to a $\hat{\sigma}$ -extension in \hat{F} (i.e. $m(E) \in \hat{\sigma}(\hat{F})$). This means that clustered framework \hat{F} does preserve all σ -extensions (under $\hat{\sigma}$ and the mapping). It might be that there are $\hat{\sigma}$ -extensions \hat{E} for \hat{F} s.t. there is no $E \in \sigma(F)$ with $m(E) = \hat{E}$. This means that \hat{E} is spurious (i.e., clustering F under m leads to a clustered AF \hat{F} that introduces “artificial” $\hat{\sigma}$ -extensions not corresponding to any σ -extensions of F). If such a case does not occur, then \hat{F} is faithful, under $\hat{\sigma}$ and w.r.t. F and σ . If \hat{F} under $\hat{\sigma}$ abstracts F under σ , and additionally is faithful w.r.t. F under σ , then we get equality under the mapping: $m(\sigma(F)) = \hat{\sigma}(\hat{F})$.

A semantics on clustered AFs $\hat{\sigma}$ is then abstracting a classical semantics σ if it holds that for every AF F and mapping m we have $m(F)$ under $\hat{\sigma}$ abstracts F under σ , or in other words, that $\hat{\sigma}$ prescribes $\hat{\sigma}$ -extensions in such a way that it is never the case that a σ -extension is “missed”.

Definition 9. A semantics $\hat{\sigma}$ on clustered AFs abstracts a semantics σ on AFs if for every set of arguments A , every mapping m on A , and every AF $F = (A, R)$ it holds that \hat{F} under $\hat{\sigma}$ abstracts F under σ for the clustered AF $\hat{F} = m(F)$.

Before we can exemplify these notions fully, we define the following semantics on clustered AFs. As the naming of each such $\hat{\sigma}$ will suggest, each is meant to abstract classical semantics σ on AFs.

Definition 10. Let $\hat{F} = (\hat{A}, \hat{R})$ be a clustered AF according to m , and $\hat{E} \subseteq \hat{A}$. We define the following semantics on clustered AFs:

- $\hat{E} \in \hat{cf}(\hat{F})$ iff for each $\hat{a}, \hat{b} \in \text{single}(\hat{E})$ we have $(\hat{a}, \hat{b}) \notin \hat{R}$,
- $\hat{E} \in \hat{adm}(\hat{F})$ iff $\hat{E} \in \hat{cf}(\hat{F})$ and if $\hat{a} \in \hat{E}$ with $(\hat{b}, \hat{a}) \in \hat{R}$ s.t. $|\hat{a}| = 1$, then there is a $\hat{c} \in \hat{E}$ with $(\hat{c}, \hat{b}) \in \hat{R}$,
- $\hat{E} \in \hat{stb}$ iff $\hat{E} \in \hat{cf}(\hat{F})$, $\hat{b} \notin \hat{E}$ implies that there is an $\hat{a} \in \hat{E}$ with $(\hat{a}, \hat{b}) \in \hat{R}$, and if \hat{E} does not attack an $\hat{a} \in \hat{E}$ then $\hat{b} \notin \hat{E}$ whenever $(\hat{a}, \hat{b}) \in \hat{R}$ and $|\hat{b}| = 1$.

That is, in a clustered AF \hat{F} , a set of clustered arguments is conflict-free (in $\hat{cf}(\hat{F})$) if it holds that there is no attack between singletons in the set. A set is admissible if it is conflict-free, and if a singleton \hat{a} inside the set is attacked, then there must be a clustered argument \hat{c} in the same set that attacks the attacker \hat{b} (however we do not require either \hat{b} or \hat{c} to be singletons). The first two conditions for stable semantics are a direct analogue to the classical stable semantics on AFs. The third condition is specific to clustered AFs: if \hat{E} does not attack an \hat{a} , which is in \hat{E} , there cannot be any singleton arguments adjacent to \hat{a} in \hat{E} .

Example 4. Going back to our running example from the introduction (Example 1 and Figure 1), it holds that \emptyset , $\{a\}$, $\{b\}$, $\{c\}$, $\{a, d\}$, $\{b, d\}$, and $\{c, e\}$ are admissible in the AF in Figure 1(a). Inspecting the clustered AF in Figure 1(b), we have the following sets in $\hat{adm}(\hat{F})$: \emptyset , $\{\hat{a}\}$, $\{\hat{a}, d\}$, and $\{\hat{a}, e\}$. While there is a “self-attack” on \hat{a} , admissibility does not restrict non-singletons (reflecting the existential nature of our abstraction: arguments inside the cluster might be defended, e.g., in the AF in Figure 1(a)). Inclusion of e in the set $\{\hat{a}, e\}$ is in a direct fashion an analogue to classical admissibility: \hat{a} defends e against d . Set $\{\hat{a}, d\}$ reflects that an argument in the cluster \hat{a} may attack another in the cluster, and then defend d against this attack (exactly the situation in Figure 1(a)). It holds that \hat{F} under \hat{adm} abstracts F under adm (applying the mapping to each admissible set in F results in \hat{adm} -extensions). No set in $\hat{adm}(\hat{F})$ is spurious (no “additional” sets than those mapped onto from $adm(F)$). This means that \hat{F} under \hat{adm} is faithful w.r.t. F under adm .

In contrast, consider the clustered AF \hat{F}' from Figure 1(c). Here we have \emptyset , $\{\hat{a}\}$, $\{d\}$, and $\{\hat{a}, d\}$ are admissible sets in \hat{F}' . Thus, the set $\{d\}$ is spurious w.r.t. F under admissibility, since there is no $E \in adm(F)$ s.t. $m'(E) = \{d\}$ (since $m'(d) = d$ this would imply that $\{d\}$ is admissible in F , a contradiction). This means that \hat{F}' under \hat{adm} is not faithful w.r.t. F under adm .

Our definitions for semantics of clustered AFs for conflict-free sets, admissible sets and stable extensions are abstracting the corresponding notions on classical AFs, as shown next. This means that whenever $m(F) = \hat{F}$ we have $m(\sigma(F)) \subseteq \hat{\sigma}(\hat{F})$ for $\sigma \in \{cf, adm, stb\}$. A remark on notation: if we specified σ to be a classical semantics on AFs, e.g., $\sigma = stb$, we mean by $\hat{\sigma}$ the semantics \hat{stb} . Throughout the paper, for proof details see the supplement at http://www.kr.tuwien.ac.at/staff/zeynep/pub/kr/SW21_supp.pdf.

Theorem 1. For $\sigma \in \{cf, adm, stb\}$ it holds that $\hat{\sigma}$ abstracts σ .

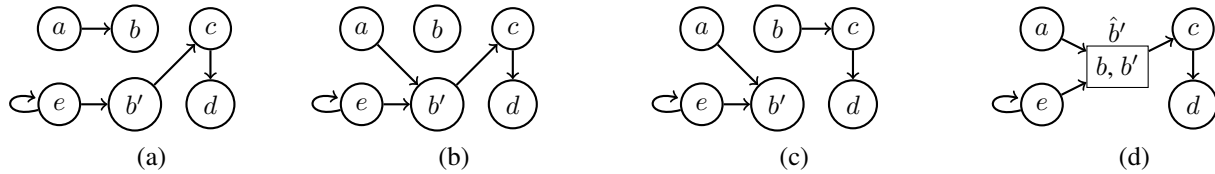


Figure 2: Three AFs F_1 (a), F_2 (b), and F_3 (c), mapped onto the same clustered AF \hat{F} (d).

As the reader might have guessed, being “abstracting” (Definition 9) is, without further properties, not very restricting: a semantics $\hat{\sigma}$ that naively collects all subsets, i.e., $\hat{\sigma}(\hat{F}) = 2^{\hat{A}}$ for each $\hat{F} = (\hat{A}, \hat{R})$, is abstracting each classical AF semantics (no σ -extension can be missed if we collect all). We next prove for conflict-free sets, admissible sets, and stable extensions, that our definitions for clustered AFs (Definition 10) are “optimal” in a formally defined sense. Concretely, for every semantics $\hat{\tau}$ on clustered AFs that abstracts σ it holds that $\hat{\sigma}(\hat{F}) \subseteq \hat{\tau}(\hat{F})$, for $\sigma \in \{cf, adm, stb\}$. In words, if a semantics abstracts σ , then it necessarily contains all sets prescribed by $\hat{\sigma}$, as defined in Definition 10. That is, if we require “being abstracting” then we need to include all sets of $\hat{\sigma}$. Together with $\hat{\sigma}$ being abstracting (Theorem 1), it follows that our definitions for these semantics on clustered AFs are as minimal as possible (permitting as few sets as possible). Even excluding one $\hat{\sigma}$ -extension leads to the semantics being not abstracting anymore.

Theorem 2. Let $\hat{F} = (\hat{A}, \hat{R})$ be a clustered AF with mapping m , and $\sigma \in \{cf, adm, stb\}$. Assume that $\hat{\tau}$ abstracts σ . It holds that $\hat{\sigma}(\hat{F}) \subseteq \hat{\tau}(\hat{F})$.

As the reader might have wondered, what about the complete, preferred, and grounded semantics? It appears that these are significantly more involved on the abstract level of clustered AFs, in particular when aiming at the same notion of optimality as exhibited in Theorem 2.

We first turn our attention to the preferred semantics. Let us consider a simple example.

Example 5. Let $F = (A, R)$ be an AF with $A = \{a, b, b'\}$ and $R = \{(a, b), (a, b')\}$. Let $m(a) = a$, and $m(b) = m(b') = \hat{b}$. We have $adm(F) = \{\emptyset, \{a\}\}$, and for the clustered AF $m(F) = \hat{F} = (\{\hat{a}, \hat{b}\}, \{(a, \hat{b})\})$ we get $adm(\hat{F}) = \{\emptyset, \{a\}, \{\hat{b}\}, \{a, \hat{b}\}\}$. Considering only subset maximal members of both semantics, we get $E = \{a\}$ as the preferred extension of F , and $\hat{E} = \{a, \hat{b}\}$ as a subset-maximal adm set of \hat{F} . However, $m(E) = \{a\} \neq \hat{E}$.

Directly from the example we can infer that defining a preferred semantics on clustered AFs is not immediate from the definition of being subset-maximal admissible.

Observation 1. The semantics $\hat{\sigma}$ on clustered AFs defined as $\hat{\sigma}(\hat{F}) = \max_{\subseteq}(adm(\hat{F}))$ for each clustered AF \hat{F} does not abstract preferred semantics.

We hypothesize that complete-based semantics (e.g., preferred, complete, and grounded) face significant barriers for defining optimal counterparts on clustered AFs. We first

give an example for grounded semantics that highlights challenges, and then argue that defining such semantics requires potentially deeper results that were already challenging to obtain on related questions.

Example 6. Consider the three AFs F_1 , F_2 , and F_3 in Figure 2(a-c). Under the mapping that clusters b and b' into \hat{b} , all three map onto the clustered AF \hat{F} in Figure 2(d). The grounded extensions are $grd(F_1) = \{a\}$, $grd(F_2) = \{a, b, c\}$, and $grd(F_3) = \{a, b, d\}$. If a semantics on clustered AFs abstracts grounded semantics, it is then required to include $\{a\}$, $\{a, \hat{b}, c\}$, and $\{a, \hat{b}, d\}$. Apart from not being a unique-status semantics (at least three sets), we face a further issue: if the cluster \hat{b} satisfies $|\hat{b}| = 2$ (i.e., exactly two arguments are mapped into \hat{b}), then $\{a, \hat{b}\}$ is spurious (i.e., cannot be a grounded extension in an AF mapped onto \hat{F}). The unattacked a and the self-attacking e necessarily attack one argument in \hat{b} (possibly different ones), excluding at least one argument in \hat{b} of being in the grounded extension. If one argument in \hat{b} is part of the grounded extension, then it cannot be attacked by either a or e . There is one attack from one of the arguments in \hat{b} to c . This attack cannot originate from the unattacked one in \hat{b} (then d is defended and included in the grounded extension). If the attack originates only from the attacked argument in \hat{b} , then c is defended. In both cases, there is no grounded extension containing exactly a and one of b and b' . However, if we expand the cluster and add a third argument b'' (i.e., $\hat{b} = \{b, b', b''\}$), then we can have an unattacked argument in \hat{b} , and an attack originating from an argument only attacked by the self-attacking e (implying then that the grounded extension contains only a and one of the cluster).

The example suggests that a semantics abstracting grounded semantics is required to include multiple potential grounded extensions, and, even more, information about cardinality (size) of clusters (not only information about singletons). This contrasts Definition 10, which could be defined using more basic concepts taken from classical AF semantics. Indeed, we hypothesize that the underlying issue is one of the related problem of realizability (Dunne et al. 2015): the proof of Theorem 2 relies on constructing AFs that map to a given clustered AF and contain specific σ -extensions. Similarly, realizability asks whether AFs exist that have specific sets of arguments as their σ -extensions. In our case, the issue appears to be connected to completeness (in AFs in Example 6, in the form of the grounded extension, but note that there is a unique preferred extension which coincides

with the grounded extension in these AFs). Realizability of complete semantics does appear to be among the most involved ones¹, and here we would need a realizability-like result in a restricted language of AFs (only those compatible with clustered AFs).

We leave optimal semantics for clustered AFs that abstract grounded, complete, or preferred semantics for future work, but remark that \hat{adm} abstracts all three semantics (however, not optimally), and moreover, in Section 7 we discuss a general method for obtaining semantics of clustered AFs. One refinement regarding \hat{adm} can be achieved: for abstracting complete, grounded, and preferred semantics, one can restrict consideration to only those sets of clustered arguments that contain all unattacked arguments. The reason is that arguments in an unattacked cluster are unattacked.

5 Properties of Clustered Semantics

We investigate properties of the clustered semantics we introduced. We start with clustered AFs that are faithful.

Proposition 3. *Let $F = (A, R)$ be an AF and $\sigma \in \{cf, adm\}$.*

- *if $\exists E \in \sigma(F)$ s.t. $E \neq \emptyset$, then $\exists \hat{R}'$ s.t. $\hat{F}' = (\{A\}, \hat{R}')$ (according to an m') under $\hat{\sigma}$ is faithful w.r.t. F under σ .*

Additionally, let m be a mapping over A , $m(F) = \hat{F} = (\hat{A}, \hat{R})$. If \hat{F} under $\hat{\sigma}$ is faithful w.r.t. F under σ , then

- *if $\hat{a} \in \hat{A} \setminus \text{single}(\hat{A})$ then there is an $E \in \sigma(F)$ with $E \cap \hat{a} \neq \emptyset$, and*
- *if no $x \in X$ is credulously accepted under σ in F and $|X| > 1$, then there is no $\hat{a} \in \hat{F}$ with $X = \hat{a}$.*

That is, a non-singleton cluster, in a clustered AF faithful to an AF under conflict-free or admissible sets, contains at least one argument that is part of one conflict-free or admissible set (i.e., contains an argument credulously accepted). The first item means that if there is at least one non-trivial σ -set (i.e., not equal to \emptyset), then one can cluster the whole set of arguments into one big cluster, and be faithful. While this item might seem to be “trivializing” clustering, it also is a starting point for an initial clustering and subsequent iterative refinements. We return to this in Section 7.1. Nevertheless, operations on clusters, like merging clusters to obtain larger ones, is not a trivial task if one desires to preserve faithfulness, as exemplified next.

Example 7. *Let $F = (A, R)$ be an AF with $A = \{a, a', b, b', c, d\}$ and $R = \{(a, a'), (b, c), (c, d)\}$. Define m such that $m(a) = m(a') = \hat{a}$, $m(b) = m(b') = \hat{b}$, $m(c) = c$, and $m(d) = d$. Let $\hat{F} = m(F)$ be the corresponding clustered AF according to m . We have \emptyset , $\{\hat{a}\}$, $\{\hat{b}\}$, $\{b'\}$, $\{a, b\}$, $\{a, b'\}$, $\{b, b'\}$, $\{b, b', d\}$, $\{a, b, d\}$, $\{b, d\}$, $\{a, b, b'\}$, and $\{a, b, b', d\}$ are admissible sets in F . Notably, there is no $E \in adm(F)$ with $c \in E$. For \hat{F} , these are the \hat{adm} sets: \emptyset , $\{\hat{a}\}$, $\{\hat{b}\}$, $\{\hat{a}, \hat{b}\}$, $\{\hat{b}, \hat{d}\}$, and $\{\hat{a}, \hat{b}, \hat{d}\}$. It follows that \hat{F} under \hat{adm} is faithful w.r.t. F under adm . However, “merging” the two clusters \hat{a} and \hat{b} results in*

¹The case for complete semantics was later added as an addendum to Dunne et al. (2015) in a technical report (Linsbichler 2018).

$m'(a) = m'(a') = m'(b) = m'(b') = \hat{a}$, $m'(c) = c$, and $m'(d) = d$. In the clustered AF \hat{F}' according to m' we have \emptyset , $\{\hat{a}\}$, $\{\hat{a}, c\}$, $\{\hat{a}, d\}$, and $\{\hat{a}, c, d\}$ are \hat{adm} in \hat{F}' (since \hat{a} is self-attacking in \hat{F}'). It follows that \hat{F}' under \hat{adm} is not faithful w.r.t. F under adm .

As an interesting corollary to Theorem 2, we also get that for each \hat{E} in $\hat{\sigma}(\hat{F})$ for a clustered AF \hat{F} according to m , it holds that there exists some classical AF F with $m(F) = \hat{F}$ such that there is an $E \in \sigma(F)$ that is mapped onto $\hat{E} = m(E)$, for $\sigma \in \{cf, adm, stb\}$. In a sense, this is a rephrasing of the corresponding theorem, highlighting that no single $\hat{\sigma}$ -extension can be removed.

Corollary 4. *Let $\hat{F} = (\hat{A}, \hat{R})$ be a clustered AF according to m , and $\sigma \in \{cf, adm, stb\}$. It holds that $\hat{E} \in \hat{\sigma}(\hat{F})$ iff there is an AF $F = (A, R)$ with $m(F) = \hat{F}$ s.t. \hat{E} is not spurious w.r.t. F under σ .*

The corollary also implies that the computational task of verifying whether for a given clustered AF \hat{F} according to an m , and a $\hat{\sigma}$ -extension \hat{E} , $\sigma \in \{cf, adm, stb\}$, there exists an AF F , with $m(F) = \hat{F}$ and an $E \in \sigma(F)$ s.t. $m(E) = \hat{E}$, is trivial. By Corollary 4, there always exists such an AF.

Nevertheless, when checking whether \hat{E} is spurious w.r.t. a given AF, the task becomes coNP-complete under admissibility.

Proposition 5. *Verifying whether a set of clustered arguments in a given clustered AF is spurious w.r.t. a given AF under admissibility is coNP-complete.*

The result suggests that checking faithfulness is not an easy task in general, computationally speaking. We can also see this from the following result. Faithfully showing credulous acceptance is defined by all $\hat{\sigma}$ -extensions containing a queried singleton clustered argument being non-spurious in a given clustered AF w.r.t. a given AF mapped to it. Or, put simply, that there are no spurious sets from which credulous acceptance could be derived.

Proposition 6. *Deciding whether a given clustered AF is faithful in showing the credulous acceptance of an argument under admissibility w.r.t. a given AF is Π_2^P -complete.*

We close our study of complexity with stating that the complexity for standard problems of checking whether a set is a σ -extensions (dubbed verification), and credulous and skeptical acceptance coincides for classical AFs and clustered AFs (hardness follows from clustered AFs containing classical AFs by mapping only to singletons, and membership directly from Definition 10).

Proposition 7. *Complexity of the problems of verification, credulous and skeptical acceptance coincides for AFs and clustered AFs under conflict-free sets, admissibility, and stable semantics.*

Finally, for this section, we note an observation on faithfulness. Assume $\hat{\sigma}$ abstracts σ . If two AFs F and F' are mapped to the same $m(F) = m(F') = \hat{F}$, and the clustered AF is faithful under $\hat{\sigma}$ to both AFs, under a σ , then both AFs have correspondences regarding their semantics (under the mapping).

Proposition 8. Assume $\hat{\sigma}$ abstracts σ . Let $F = (A, R)$ and $F' = (A, R')$ be AFs, m a mapping over A , and $m(F) = m(F') = \hat{F} = (\hat{A}, \hat{R})$ a clustered AF according to m . Let $S = \text{single}(\hat{A})$. If \hat{F} under $\hat{\sigma}$ is faithful w.r.t. both F and F' under σ , then

- $m(\sigma(F)) = m(\sigma(F'))$ and
- $\{E \cap S \mid E \in \sigma(F)\} = \{E' \cap S \mid E' \in \sigma(F')\}$.

Since these AFs share the same set of arguments, this means that the associated attack structures induce the same σ -extensions, under the mapping.

6 Clustering for Reasoning Tasks

We discuss the use of clustering for the tasks of credulous and skeptical reasoning. We begin with the positive case, i.e., we look at an argument a credulously accepted under σ in an AF $F = (A, R)$. When considering clustered AFs, for this case faithfulness appears as a strict requirement: if $m(F) = \hat{F}$ is a clustered AF according to m , and if $a \in \hat{E}$ for some $\hat{E} \in \hat{\sigma}(\hat{F})$, then \hat{E} might be spurious. In this case we do not have a witness for credulous acceptance of a (in fact a might not be credulously accepted in the original AF).

Proposition 9. Let $F = (A, R)$ be an AF, m a mapping on A , and $m(F) = \hat{F} = (\hat{A}, \hat{R})$ under $\hat{\sigma}$ faithful w.r.t. F under σ . For an $a \in \text{single}(\hat{A})$ it holds that $a \in A$ is credulously accepted in F under σ iff a is credulously accepted in \hat{F} under $\hat{\sigma}$.

Strictly speaking, faithfulness is only required in a restricted form: those $\hat{\sigma}$ -extensions that contain a should not be spurious. Note that if $m(a) = \hat{a}$ where \hat{a} is not a singleton, then the existence of $\hat{a} \in \hat{E}$ might not be a witness for credulous acceptance of a , but for some other argument a' mapped to \hat{a} .

For the negative case, say a is not credulously accepted under admissibility. Interestingly, abstraction directly gives indication of this acceptance status.

Proposition 10. Assume $\hat{\sigma}$ abstracts σ , and let $\hat{F} = (\hat{A}, \hat{R})$ be a clustered AF according to m , and $\hat{a} \in \hat{A}$. If there is no $\hat{E} \in \hat{\sigma}(\hat{F})$ with $\hat{a} \in \hat{E}$, then for all F such that $m(F) = \hat{F}$, it holds that there is no $E \in \sigma(F)$ with $E \cap \hat{a} \neq \emptyset$.

This statement follows from the definition of a semantics being abstracting (Definition 9). Note that for $\sigma = \text{adm}$ we can directly infer that then \hat{a} is a singleton (see Proposition 3). In more words, if $m(F) = \hat{F}$ for some F and mapping m , and we find \hat{a} is not credulously accepted, under $\hat{\sigma}$ in $m(F)$, then all arguments clustered into \hat{a} are not credulously accepted under σ in the original F . This holds independently of \hat{F} being faithful (follows from abstraction). For stable semantics, if there are no stable extensions in the clustered AF, then there are no stable extensions in the AFs mapped to the clustered one. More formally, we have the following (which is a consequence of $m(\sigma(F)) \subseteq \hat{\sigma}(\hat{F})$).

Proposition 11. Assume $\hat{\sigma}$ abstracts σ , and let $\hat{F} = (\hat{A}, \hat{R})$ be a clustered AF according to m . If $\hat{\sigma}(\hat{F}) = \emptyset$, then for all F such that $m(F) = \hat{F}$, we have $\sigma(F) = \emptyset$.

Example 8. Consider the somewhat more involved AF in Figure 3(a). Let us focus on non-acceptance under the credulous view. The AF has no stable extensions, and c is not credulously accepted. The reasons can be seen with clustering. The clustered AF in (b) has no stable extensions (no $\hat{\sigma}$ -extensions): we need to include e (for attacking f), and then we can include neither c nor d . This leaves only the cluster for inclusion in a potential stable set, but then d is neither attacked nor inside the set, a contradiction to stability. In particular, it holds that any AF mapped to this clustered AF does not have any stable extensions. This clustered AF provides the underlying reason: the particular odd-cycle and self-attacking f prevent stability. It is of no importance how the attacks compatible with the cluster exactly look like (as long as they are compatible).

For credulous non-acceptance under admissibility of c , consider clustered AF in Figure 3(c). To defend c against attacks from e or the cluster, we need d , but d and c are conflicting. However, this clustered AF has a stable extension: d together with the cluster attacks all other arguments (and conflict-freeness is weaker on clusters: we restrict only attacks regarding singletons). While this stable extension is spurious w.r.t. the AF on the left side, there exists an AF mapped to this clustered AF with such a stable extension. Consider, e.g., to have d not attack a , a attack c , and d attacks all b, f, g , and h , and b self-attacking.

Reasoning skeptically is, naturally, dual to credulous acceptance. If a singleton clustered argument is skeptically accepted in a clustered AF, then for all AFs mapped onto this one, we get that this argument is skeptically accepted, as well. Note that being a singleton is required.

Proposition 12. Assume $\hat{\sigma}$ abstracts σ , and let $\hat{F} = (\hat{A}, \hat{R})$ be a clustered AF according to m . If there exists $a \in \text{single}(\hat{A})$ such that $a \in \hat{E}$ for all $\hat{E} \in \hat{\sigma}(\hat{F})$, then for all F such that $m(F) = \hat{F}$, a is skeptically accepted under σ in F .

7 Clustered AFs with ASP Abstraction

Recently, a notion of abstraction (Saribatur and Eiter 2018; Saribatur, Schüller, and Eiter 2019) inspired by Clarke et al. (2003) has been introduced to answer set programming (ASP) as a means for simplifying the ASP program and its vocabulary while ensuring that the original answer sets are preserved in the abstract program, with the potential of introducing spuriousness.

In this section, we show how the ASP abstraction method can be used to obtain and reason over clustered AFs.

Abstraction in ASP We first recall the basics of ASP. A logic program Π is a finite set of rules of the form

$$\alpha_0 \leftarrow \alpha_1, \dots, \alpha_m, \text{not } \alpha_{m+1}, \dots, \text{not } \alpha_n, \quad 0 \leq m \leq n.$$

where each α_i is an atom and *not* is default negation. We also write r as $\alpha_0 \leftarrow B(r)$, such that $H(r) = \alpha_0$ denotes the head or as $H(r) \leftarrow B^+(r), \text{not } B^-(r)$, where $B^+(r) = \{\alpha_1, \dots, \alpha_m\}$ is the positive body and $B^-(r) = \{\alpha_{m+1}, \dots, \alpha_n\}$ is the negative body of r . We may omit r from $B(r), B^+(r)$ etc. if r is clear. A rule r is a constraint, if

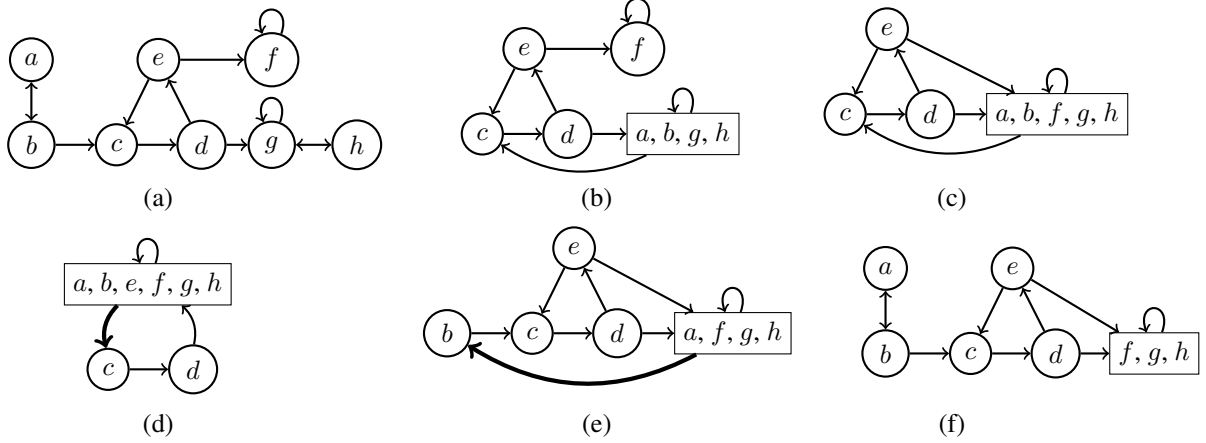


Figure 3: AF (a), two clustered AFs for no stable extensions (b) and c not credulously accepted under admissibility (c), and three clustered AFs (d-e-f) for witnessing credulous acceptance of d , where (e) and (f) are refinements of (d) and (e), respectively, showing more details for the (bold) attacks of the cluster.

α_0 is falsity (\perp , then omitted) and a *fact*, if $n=0$ and no variable occurs in r . A rule is *ground*, if all atoms occurring in it are ground. *Choice rules* are syntactic sugar of the form $\{a\} \leftarrow B$, which stands for the rules $a \leftarrow B, not \bar{a}$ and $\bar{a} \leftarrow B, not a$, where \bar{a} is a new atom. An *interpretation* I is a subset of the Herbrand base, and it is an *answer set* of Π if it is a minimal model of the reduct $\Pi^I = \{r \in ground(\Pi) \mid I \models B(r)\}$ (Faber, Leone, and Pfeifer 2004).

In the well-known ASP encodings for AFs and AF semantics (Egly, Gaggl, and Woltran 2010), the graph structure of an AF $F = (A, R)$ is encoded by $\pi_F = \{\mathbf{arg}(a). \mid a \in A\} \cup \{\mathbf{att}(a, b). \mid (a, b) \in R\}$. We encode credulous acceptance of $a \in A$ by constraint $\leftarrow not \mathbf{in}(a)$. That is, the corresponding ASP has an answer set iff the argument is credulously accepted. Admissible semantics can be encoded as follows (Egly, Gaggl, and Woltran 2010).

$$\begin{aligned} \{\mathbf{in}(X)\} &\leftarrow \mathbf{arg}(X). \\ &\leftarrow \mathbf{in}(X), \mathbf{in}(Y), \mathbf{att}(X, Y). \\ \mathbf{def}(X) &\leftarrow \mathbf{in}(Y), \mathbf{att}(Y, X). \\ &\leftarrow \mathbf{in}(X), \mathbf{att}(Y, X), not \mathbf{def}(Y). \end{aligned}$$

A generic notion of abstraction in ASP is as follows.

Definition 11 (cf. Saribatur and Eiter, 2018). *Given two programs Π and Π' on sets \mathcal{A} and \mathcal{A}' of ground atoms, respectively, where $|\mathcal{A}| \geq |\mathcal{A}'|$, and a mapping $m : \mathcal{A} \rightarrow \mathcal{A}'$, Π' is an abstraction of Π w.r.t. m , if for every answer set I of Π , $I' = \{m(\alpha) \mid \alpha \in I\}$ is an answer set of Π' .*

The propositional view of this notion is then lifted to a first-order view in Saribatur, Schüller, and Eiter (2019). The abstraction mapping is defined over the Herbrand universe of Π , called *domain*, by merging the constants.

Definition 12 (cf. Saribatur, Schüller, and Eiter, 2019). *Given a domain D of Π , a (domain abstraction) mapping is a function $m : D \rightarrow \hat{D}$ for a set \hat{D} (the abstracted domain) with $|\hat{D}| \leq |D|$.*

A domain abstraction mapping divides D into clusters of elements. Abstracting the elements in the Herbrand universe induces an abstraction of the Herbrand base. Each domain abstraction mapping m naturally extends to ground atoms $\alpha = p(v_1, \dots, v_n)$ by $m(\alpha) = p(m(v_1), \dots, m(v_n))$.

Given a (non-ground) program Π and an induced mapping $m : \mathcal{A} \rightarrow \hat{\mathcal{A}}$ from the Herbrand base \mathcal{A} of Π to $\hat{\mathcal{A}} = m(\mathcal{A}) = \{m(\alpha) \mid \alpha \in \mathcal{A}\}$, Saribatur, Schüller, and Eiter (2019) present a method to construct an abstract (non-ground) program Π' that achieves over-approximation over the answer sets as in Definition 11. Here, we make use of this method, and refer the reader for details to the referenced paper.

For an AF $F = (A, R)$, given an argument mapping m over A , we obtain the abstraction of π_F for $m(F) = \hat{F} = (\hat{A}, \hat{R})$ as $\pi_{\hat{F}} = \{\mathbf{arg}(\hat{a}). \mid \hat{a} \in \hat{A}\} \cup \{\mathbf{att}(\hat{a}, \hat{b}). \mid (\hat{a}, \hat{b}) \in \hat{R}\} \cup \{\mathbf{single}(\hat{a}) \mid \hat{a} \in \hat{A}, |\hat{A}| = 1\}$. The abstracted encoding of admissible sets becomes as shown next.

$$\begin{aligned} \{\mathbf{in}(X)\} &\leftarrow \mathbf{arg}(X). \\ &\leftarrow \mathbf{in}(X), \mathbf{in}(Y), \mathbf{att}(X, Y), \mathbf{single}(X), \\ &\quad \mathbf{single}(Y). \\ \mathbf{def}(X) &\leftarrow \mathbf{in}(Y), \mathbf{att}(Y, X), \mathbf{single}(Y). \\ \{\mathbf{def}(X)\} &\leftarrow \mathbf{in}(Y), \mathbf{att}(Y, X), not \mathbf{single}(Y). \\ &\leftarrow \mathbf{in}(X), \mathbf{att}(Y, X), not \mathbf{def}(Y), \mathbf{single}(X). \end{aligned}$$

Observe that this abstract admissible semantics encoding exactly matches $\hat{a}dm$, while also containing $\hat{c}f$ in the first two rules. Note that using the method by Saribatur, Schüller, and Eiter (2019), we can abstract on an ASP encoding of an AF semantics to obtain an abstract semantics, automatically. Though, as discussed before, optimality of the abstract semantics might not be guaranteed, e.g., for complete.

7.1 Obtaining Explanatory Clustered AFs

In Saribatur, Schüller, and Eiter (2019), an abstraction and refinement methodology (inspired by Clarke et al. (2003))

is introduced with prototype tools, that allows to automatically compute abstractions by starting with an initial coarse abstraction and refining it whenever a spurious answer set is encountered. This process continues until a non-spurious answer set is encountered or the unsatisfiability is observed. For the non-acceptance case, we can, with this tool, automatically find abstraction mappings to obtain non-acceptance at the abstract level as described in Example 8: starting with a highly coarse abstraction of clustering all the nodes into one, and automatically refining whenever spuriousness is encountered. Note that for achieving the abstraction in Figure 3(b), the abstract stable encoding obtained with the method by Saribatur, Schüller, and Eiter (2019), which is able to check the first two conditions of $\hat{s}b$, can also be used. As discussed in Example 8, the obtained abstractions give an understanding for the non-acceptance cases.

For our purposes on reasoning over clustered AFs for the acceptance cases, achieving faithful abstractions shows to be important. For this, we modified the tool to compute faithful abstractions, by checking all of the answer sets of the computed abstract program, and refining the mapping whenever the faithfulness is not achieved.

However, for credulous acceptance of arguments under admissibility, clustering all arguments into one large cluster also results in a faithful clustered AF (as indicated by Proposition 3). Consider the graph in Figure 3(a) and the credulous acceptance of d . Clustering all arguments into one results in faithfulness w.r.t the credulous acceptance of d . However, such a large cluster would not contain any explanatory details on the acceptance of d in the original AF. To obtain further details, one can ask for the refinement of this clustering to one where d is mapped to a singleton, and given this refinement, start the search for a new faithful clustered AF. Figure 3(d) shows a mapping which shows more details. Here, we can infer that there exists an argument in the cluster defending d which is not in conflict with d . If further details on the clustered AF are required, one can ask for the refinement of this clustering, by making (all) attackers of c singleton, which gives us Figure 3(e). Further such refinement would result in Figure 3(f). Such a guidance by the user on the refinement to obtain a new faithful clustered AF would help the user in understanding the reason for the credulous acceptance of d , while the irrelevant parts of the AF still remain abstracted.

The tool with the search for faithful abstractions and the user guided refinement can be found at <http://www.kr.tuwien.ac.at/research/systems/abstraction/>.

8 Related Work

Broadly speaking, abstraction is used in many areas for state space reduction: e.g., in model checking (Clarke, Grumberg, and Long 1994), in multi-agent systems (Cohen et al. 2009), and in planning (Edelkamp 2001; Helmert, Haslum, and Hoffmann 2007; Seipp and Helmert 2013). The well-known counterexample guided abstraction refinement (Clarke et al. 2003) approach automatically refines abstractions.

Clustering on AFs, in rather different directions, is studied using semantic similarity measures of arguments (Block et

al. 2019), for classification (Gómez and Chesñevar 2003), and for computing reasoning tasks efficiently (Doutre, Lafages, and Lagasque-Schiex 2019). Block argumentation (Arisaka, Santini, and Bistarelli 2019) and fibring argument frames (Gabbay 2009) can be seen as a kind of “clustering” in the sense that an argument may stand for whole AFs. In Boella, Kaci, and van der Torre (2009), properties of abstraction by removal of arguments and attacks are studied. To our knowledge, the closest approach to ours is presented in a technical report (Arisaka and Dauphin 2018), which, for the treatment of cycles, proposed a different kind of abstraction motivated by abstract interpretation (Cousot and Cousot 1977) based on monotonic functions over ordered sets, in particular for sharpening extensions, which induce different acceptance statuses.

Our work is connected to the notion of realizability (Dunne et al. 2015) as discussed before. Through an observation like in Proposition 8, our work can be related to works studying replacements of parts of an AF (Baroni et al. 2014; Dvořák et al. 2019) which investigate, e.g., simplifications (modifications) that preserve equivalences relativized to certain arguments.

For simplification, works that obtain subframeworks (Fan and Toni 2015; Saribatur, Wallner, and Woltran 2020; Niskanen and Järvisalo 2020; Ulbricht and Wallner 2021) and removal of arguments and attacks, e.g., in abduction (Sakama 2018), relate to our clustering approach, but differ in not considering clusterings. Incomplete AFs (Baumeister et al. 2018) are AFs with some parts being uncertain. When considering all arguments in clusters, and making possible compatible attacks uncertain, an AF mapped to a clustered AF can be seen as a completion in the framework of incomplete AFs, however the set of completions does not directly correspond to all AFs mapped to the same clustered AF (in our approach existential existence of attacks must be respected).

9 Conclusions

We introduced existential abstraction on argumentation frameworks, via clustering arguments. We presented semantics for the clustered AFs which abstract corresponding semantics for classical AFs, showed properties, complexity results, and an automated way of deriving clusterings and reasoning on clustered AFs. We view clustering (existential abstraction) as part of a larger set of methodological tools to simplify AFs, for enhancing explanation capabilities.

Our clustering is applied on AFs, in particular by focusing on existential abstraction according to the attack structure and argumentation semantics as vehicles driving the clustering. Clustering of arguments in this way is not restricted to arguments without internal structure: faithful clusterings of structured arguments (if following the prominent approach of using AFs as reasoning engines) preserve argumentation semantics still when including internal structure of arguments. Moreover, we think that when taking internal structures additionally into account, one can have more fine-grained control on the clustering, e.g., by preferring clusters whose arguments share structural similarities, which we view as a promising direction for future research that builds upon our work.

Acknowledgements

This work was funded by the Austrian Science Fund (FWF): P30168-N31.

References

- Amgoud, L.; Besnard, P.; and Vesic, S. 2014. Equivalence in logic-based argumentation. *Journal of Applied Non-Classical Logics* 24(3):181–208.
- Arisaka, R., and Dauphin, J. 2018. Abstractly interpreting argumentation frameworks for sharpening extensions. *CoRR* abs/1802.01526.
- Arisaka, R.; Santini, F.; and Bistarelli, S. 2019. Block argumentation. In *Proc. PRIMA*, volume 11873 of *Lecture Notes in Computer Science*, 618–626. Springer.
- 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 Magazine* 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. *Artificial Intelligence* 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. *The Knowledge Engineering Review* 26(4):365–410.
- Baumeister, D.; Neugebauer, D.; Rothe, J.; and Schadrack, H. 2018. Verification in incomplete argumentation frameworks. *Artificial Intelligence* 264:1–26.
- Bistarelli, S.; Kotthoff, L.; Santini, F.; and Taticchi, C. 2020. A first overview of ICCMA’19. In *Proc. AIXIA*, volume 2777 of *CEUR Workshop Proceedings*, 90–102. CEUR-WS.org.
- Block, K.; Trumm, S.; Sahitaj, P.; Ollinger, S.; and Bergmann, R. 2019. Clustering of argument graphs using semantic similarity measures. In *Proc. KI*, volume 11793 of *Lecture Notes in Computer Science*, 101–114. Springer.
- Boella, G.; Kaci, S.; and van der Torre, L. W. N. 2009. Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In *Proc. ECSQARU*, volume 5590 of *Lecture Notes in Computer Science*, 107–118. Springer.
- Caminada, M. 2018. Rationality postulates: Applying argumentation theory for non-monotonic reasoning. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 15.
- 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.
- Clarke, E. M.; Grumberg, O.; and Long, D. E. 1994. Model checking and abstraction. *ACM transactions on Programming Languages and Systems (TOPLAS)* 16(5):1512–1542.
- Cohen, M.; Dam, M.; Lomuscio, A.; and Russo, F. 2009. Abstraction in model checking multi-agent systems. In *Proc. AAMAS, Volume 2*, 945–952. IFAAMAS.
- Cousot, P., and Cousot, R. 1977. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. POPL*, 238–252. ACM.
- Doutre, S.; Lafages, M.; and Lagasquie-Schiex, M. 2019. A distributed and clustering-based algorithm for the enumeration problem in abstract argumentation. In *Proc. PRIMA*, volume 11873 of *Lecture Notes in Computer Science*, 87–105. Springer.
- Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321–358.
- Dunne, P. E.; Dvořák, W.; Linsbichler, T.; and Woltran, S. 2015. Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence* 228:153–178.
- Dvořák, W.; Jarvisalo, M.; Linsbichler, T.; Niskanen, A.; and Woltran, S. 2019. Preprocessing argumentation frameworks via replacement patterns. In *Proc. JELIA*, volume 11468 of *Lecture Notes in Computer Science*, 116–132. Springer.
- Edelkamp, S. 2001. Planning with pattern databases. In *Proc. ECP*, 13–24.
- Egly, U.; Gaggl, S. A.; and Woltran, S. 2010. Answer-set programming encodings for argumentation frameworks. *Argument & Computation* 1(2):147–177.
- Faber, W.; Leone, N.; and Pfeifer, G. 2004. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *Proc. JELIA*, volume 3229 of *Lecture Notes in Computer Science*, 200–212. Springer.
- Fan, X., and Toni, F. 2015. On computing explanations in argumentation. In *Proc. AAAI*, 1496–1502. AAAI Press.
- Gabbay, D. M. 2009. Fibring argumentation frames. *Studia Logica* 93(2-3):231–295.
- Gaggl, S. A.; Linsbichler, T.; Maratea, M.; and Woltran, S. 2020. Design and results of the second international competition on computational models of argumentation. *Artificial Intelligence* 279.
- Gómez, S. A., and Chesñevar, C. I. 2003. Combining argumentation and clustering techniques in pattern classification problems. In *Proc. Argentinean Congress in Computer Science*, 601–612.
- Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In *Proc. ICAPS*, 176–183. AAAI.
- Lehtonen, T.; Wallner, J. P.; and Jarvisalo, M. 2017. From structured to abstract argumentation: Assumption-based acceptance via AF reasoning. In *Proc. ECSQARU*, volume 10369 of *Lecture Notes in Computer Science*, 57–68. Springer.
- Linsbichler, T. 2018. Characteristics of multiple viewpoints in abstract argumentation under complete semantics. Technical Report DBAI-TR-2018-113, Databases and Artificial

Intelligence Group, Institute of Logic and Computation, TU Wien.

Niskanen, A., and Järvisalo, M. 2020. Smallest explanations and diagnoses of rejection in abstract argumentation. In *Proc. KR*, 667–671.

Sakama, C. 2018. Abduction in argumentation frameworks. *Journal of Applied Non-Classical Logics* 28(2-3):218–239.

Saribatur, Z. G., and Eiter, T. 2018. Omission-based abstraction for answer set programs. In *Proc. KR*, 42–51. AAAI Press.

Saribatur, Z. G.; Schüller, P.; and Eiter, T. 2019. Abstraction for non-ground answer set programs. In *Proc. JELIA*, volume 11468 of *Lecture Notes in Computer Science*, 576–592. Springer.

Saribatur, Z. G.; Wallner, J. P.; and Woltran, S. 2020. Explaining non-acceptability in abstract argumentation. In *Proc. ECAI*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, 881–888. IOS Press.

Seipp, J., and Helmert, M. 2013. Counterexample-guided cartesian abstraction refinement. In *Proc. ICAPS*. AAAI.

Thimm, M., and Villata, S. 2017. The first international competition on computational models of argumentation: Results and analysis. *Artificial Intelligence* 252:267–294.

Ulbricht, M., and Wallner, J. P. 2021. Strong explanations in abstract argumentation. In *Proc. AAIL*, 6496–6504. AAAI Press.

Yun, B.; Vesic, S.; and Croitoru, M. 2018. Toward a more efficient generation of structured argumentation graphs. In *Proc. COMMA*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, 205–212. IOS Press.