On Dynamics in Structured Argumentation Formalisms

Anna Rapberger[1] and **Markus Ulbricht**[2]
[1]TU Wien, Institute of Logic and Computation
[2]Leipzig University, Department of Computer Science
[1]arapberg@dbai.tuwien.ac.at [2]mulbricht@informatik.uni-leipzig.de

Abstract

In this paper we contribute to the investigation of dynamics in assumption-based argumentation (ABA) and investigate situations where a given knowledge base undergoes certain changes. We show that two frequently investigated problems, namely enforcement of a given target atom and deciding strong equivalence of two given ABA frameworks, are intractable in general. Interestingly, these problems are both tractable for abstract argumentation frameworks (AFs) which admit a close correspondence to ABA by constructing semantics-preserving instances. Inspired by this observation, we search for tractable fragments for ABA frameworks by means of the instantiated AFs. We argue that the usual instantiation procedure is not suitable for the investigation of dynamic scenarios since too much information is lost when constructing the AF. We thus consider an extension of AFs, called cvAFs, equipping arguments with conclusions and vulnerabilities in order to better anticipate their role after the underlying knowledge base is extended. We investigate enforcement and strong equivalence for cvAFs and present syntactic conditions to decide them. We show that the correspondence between cvAFs and ABA frameworks is close enough to capture ABA also in dynamic scenarios. This yields the desired tractable ABA fragment. We furthermore discuss consequences for the corresponding problems for logic programs.

1 Introduction

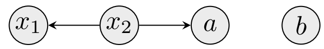
A currently highly relevant area of research in knowledge representation and reasoning is the investigation of dynamical environments, i.e., knowledge bases that change over time (Gabbay et al. 2021). Considering the inherently dynamic nature of argumentation it is not surprising that researchers in the field of formal argumentation have taken up this topic in various ways. In the area of abstract argumentation (Dung 1995) where argument acceptance is decided solely by looking at conflicts between arguments, several problems have been investigated.

Among the most prominent problems in this line of research is *strong equivalence*: Given a knowledge base $\mathcal{K}$, is it possible to replace a subset $\mathcal{H}$ of $\mathcal{K}$ by an equivalent one, say $\mathcal{H}'$, without changing the meaning of $\mathcal{K}$? Within the KR community it is folklore that this is usually not the case when considering non-monotonic formalisms. Driven by this observation, the notion of strong equivalence has been proposed, developed and investigated in various contexts (Lifschitz, Pearce, and Valverde 2001; Oikarinen and Woltran 2011). While strong equivalence is about comparing the behavior of different knowledge bases, the *enforcement problem* (Baumann 2012b; Wallner, Niskanen, and Järvisalo 2017; Doutre and Mailly 2018; Borg and Bex 2021) deals with manipulating a single one in order to ensure a certain outcome. Research concerned with this issue contributes to predict conceivable future scenarios and possible outcomes of a debate and can serve as a guidance when trying to defend a certain point of view. Both strong equivalence and enforcement have received increasing attention in the realm of abstract argumentation (Baumann et al. 2021). There are, however, only few studies on the aforementioned problems in structured argumentation; we refer the reader to Section 8 for pointers to related work.

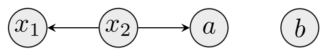
In this paper, we study the enforcement and strong equivalence problem for structured argumentation with main focus on assumption-based argumentation (ABA) (Bondarenko, Toni, and Kowalski 1993). While for abstract argumentation, deciding strong equivalence as well as the basic argument enforcement (Baumann and Brewka 2010) is tractable, it is not clear whether, and if so, how these results survive the transition to structured argumentation formalisms. At first glance it seems that we can rely on well-established methods: viewing arguments as abstract entities, we can represent instances of structured argumentation formalisms as abstract Dung-style argumentation frameworks (AFs); cf. instantiations for ABA (Caminada et al. 2015a) or logic-based argumentation (Gorogiannis and Hunter 2011). A similar procedure also exists for logic programs (LPs) (Dung 1995; Caminada et al. 2015b). Such *instantiation procedures* provide a unifying framework to study properties that are common to a large class of non-monotonic formalisms; and one would expect that they can be utilized to prove tractability or identify tractable fragments of the respective problems in the original formalisms – it is for instance well-known that deciding strong equivalence in the closely related realm of LPs is intractable (Lifschitz, Pearce, and Valverde 2001); here, we would hope that transferring the results from abstract argumentation will be helpful to identify an LP fragment for which deciding strong equivalence is tractable.

A closer inspection of the aforementioned instantiation procedures however reveals a certain drawback that becomes apparent when moving from static to dynamic scenarios.

Example 1.1. *We consider an instantiation of an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ with assumptions $\mathcal{A} = \{a, b\}$, their contraries $\overline{a}$ and $\overline{b}$, resp., and rules $r_1 = (p \leftarrow a.)$ and $r_2 = (\overline{a} \leftarrow b.)$. We obtain the associated AF F_D as follows (cf. Section 2): each assumption a, b yields a corresponding argument and each rule r_i yields an argument x_i. Attacks depend on the conclusion of the attacking argument, e.g., x_2 attacks x_1 because $\overline{a}$ is the contrary of a.*

F_D :


It turns out that we have abstracted away critical information: The rule r_2 can be disabled by adding a rule with conclusion $\overline{b}$, e.g., the fact "$\overline{b} \leftarrow .$"; this is however not reflected in F_D. To illustrate this, let us consider an adjusted version D' of D by replacing r_2 with rule "$r_2' : \overline{a} \leftarrow$", i.e., $\overline{a}$ can be considered as fact. The instantiation yields the same AF:

$F_{D'}$:


The instantiated AFs do not carry sufficient information to investigate dynamics. Consider the following questions:

- *Is it possible to accept assumption a by adding suitable rules? The answer is "yes" in D, but "no" in D'. This information cannot be extracted from $F_D = F_{D'}$.*
- *What are the stable models after adding the fact "$\overline{b} \leftarrow .$"? In D, $\{a\}$ is stable while in D', we obtain $\{b\}$. We cannot judge the situation correctly by comparing F_D and $F_{D'}$.*
- *More generally, are D and D' strongly equivalent? The answer is clearly "no" when inspecting D and D' but again we cannot tell by comparing their associated AFs.*

In all of these questions, the missing peace of information is that x_2 has a hidden weakness *$\overline{b}$ in F_D but not in $F_{D'}$. It is thus impossible to attack x_2 in $F_{D'}$ whereas in F_D, x_2 can be attacked by an argument with conclusion $\overline{b}$.*

As this example shows, the minimal generalization to tailor AFs suitable for dynamic settings consists of two aspects: (i) the *conclusion* and (ii) the *vulnerabilities* of an argument. The latter describes all possibilities to attack an argument, i.e., it contains conclusions of all potential attackers. This means that for an argument $S \vdash_R p$ in the spirit of ABA, (i.e., atom p is derivable from assumptions S via rules R) the vulnerabilities are the contraries of the assumptions in S while p is the argument's conclusion. A potential weakness of the logic-based argument $(\{\alpha, \alpha \rightarrow \beta\}, \beta)$ is the sentence $\neg\alpha$; its conclusion is β. Considering ASPIC (Modgil and Prakken 2017), also a rule can be a vulnerability: an argument $B : q \Rightarrow p$ with defeasible rule $d_1 : q \Rightarrow p$ can be attacked by an argument with conclusion $\neg d_1$.

In this paper, we consider a generalization of AFs by augmenting arguments with *vulnerabilities* and a *conclusion*. This allows us to identify a fragment of ABA for which deciding enforcement and strong equivalence becomes tractable. Our main contributions are as follows:

- We study enforcement and strong equivalence for ABA. We show that, as anticipated, both problems lie on the first level of the polynomial hierarchy and are thus intractable, in contrast to their counterparts in abstract argumentation.
- We present our novel formalism: *conclusion and vulnerability augmented AFs (cvAFs)*. We show that they are a faithful generalization of standard instantiation procedures and discuss their relation to ABA.
- We present cvAF characterization results for argument and conclusion enforcement and show that strong equivalence can be characterized by so-called kernels. Our results show that both problems are tractable for cvAFs.
- We identify a tractable fragment for ABA by means of our cvAF enforcement and strong equivalence results.
- To demonstrate the flexibility of our approach, we also transfer our results to LPs and identify a fragment for which enforcement and strong equivalence is tractable.

2 Background

Abstract Argumentation. We fix a non-finite background set $\mathcal{U}$. An argumentation framework (AF) (Dung 1995) is a directed graph $F = (A, R)$ where $A \subseteq \mathcal{U}$ represents a set of arguments and $R \subseteq A \times A$ models *attacks* between them. For a set $E \subseteq A$, we let $E_F^+ = \{x \in A \mid \exists y \in E, (y, x) \in R\}$; also, E is *conflict-free* in F iff for no $x, y \in E$, $(x, y) \in R$. E *defends* an argument x if E attacks each attacker of x. A conflict-free set E is *admissible* in F ($E \in ad(F)$) iff it defends all its elements. A *semantics* is a function $\sigma : \mathcal{F} \rightarrow 2^{2^{\mathcal{U}}}$ with $F \mapsto \sigma(F) \subseteq 2^A$; each $E \in \sigma(F)$ is called a σ-*extensions*. Here we consider so-called *complete*, *grounded*, *preferred*, and *stable* semantics (abbr. co, gr, pr, stb).

Definition 2.1. *Let $F = (A, R)$ be an AF and $E \in ad(F)$.*

- *$E \in co(F)$ iff E contains all arguments it defends;*
- *$E \in gr(F)$ iff E is $\subseteq$-minimal in $co(F)$;*
- *$E \in pr(F)$ iff E is $\subseteq$-maximal in $co(F)$;*
- *$E \in stb(F)$ iff $E^+ = A \setminus E$.*

Assumption-based Argumentation. We assume a deductive system $(\mathcal{L}, \mathcal{R})$, where $\mathcal{L}$ is a formal language and $\mathcal{R}$ is a set of inference rules over $\mathcal{L}$. A rule $r \in \mathcal{R}$ has the form $a_0 \leftarrow a_1, \ldots, a_n$, $a_i \in \mathcal{L}$; $head(r) = a_0$ is the head and $body(r) = \{a_1, \ldots, a_n\}$ is the (possibly empty) body of r.

Definition 2.2. *An ABA framework is a tuple $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, where $(\mathcal{L}, \mathcal{R})$ is a deductive system, $\mathcal{A} \subseteq \mathcal{L}$ a non-empty set of assumptions, and a contrary function $^- : \mathcal{A} \rightarrow \mathcal{L}$.*

In this work, we focus on frameworks which are *flat*, i.e., $head(r) \notin \mathcal{A}$ for each rule $r \in \mathcal{R}$, and *finite*, i.e., $\mathcal{L}$, $\mathcal{R}$, $\mathcal{A}$ are finite; also, each rule is stated explicitly (given as input).

A sentence $p \in \mathcal{L}$ is tree-derivable from assumptions $S \subseteq \mathcal{A}$ and rules $R \subseteq \mathcal{R}$, denoted by $S \vdash_R p$, if there is a finite rooted labeled tree T such that the root is labeled with p, the set of labels for the leaves of T is equal to S or $S \cup \{\top\}$, and there is a surjective mapping from the set of internal nodes to R satisfying for each internal node v there is a rule $r \in R$ such that v labelled with $head(r)$ and the set of all successor nodes corresponds to $body(r)$ or $\top$ if $body(r) = \emptyset$.

A set of assumptions S *attacks* a set of assumptions T if there is $S' \subseteq S$, $R \subseteq \mathcal{R}$, such that $S \vdash_R \overline{a}$ for some $a \in T$. S is conflict-free if it does not attack itself; S is admissible if it defends itself. We next recall grounded, complete, preferred, and stable ABA semantics (abbr. gr, co, pr, stb).

Definition 2.3. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework. Further, let $S \subseteq \mathcal{A}$ be admissible in D.*

- *$S \in co(D)$ iff S contains every assumption set it defends;*
- *$S \in gr(D)$ iff S is $\subseteq$-minimal in $co(D)$;*
- *$S \in pr(D)$ iff S is $\subseteq$-maximal in $co(D)$;*
- *$S \in stb(D)$ iff S attacks each $\{x\} \subseteq \mathcal{A} \setminus S$.*

By $Th_D(S) = \{p \mid \exists S' \subseteq S : S' \vdash_R p\}$ we denote the set of all conclusions derivable from an assumption-set S in an ABA D. Observe that $S \subseteq Th_D(S)$ since per definition, each assumption $a \in \mathcal{A}$ is derivable from $\{a\} \vdash_\emptyset a$. We call $Th_D(S) \setminus S$ the set of *proper* conclusions of X. We say that an assumption $a \in \mathcal{A}$ (atom $p \in \mathcal{L}$) is *credulously accepted* wrt a semantics σ in an ABA D iff there is some $S \in \sigma(D)$ with $a \in S$ ($p \in Th_D(S)$, respectively).

Definition 2.4. *The associated AF $F_D = (A, R)$ of an ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ is given by $A = \{S \vdash p \mid \exists R \subseteq \mathcal{R} : S \vdash_R p\}$ and attack relation $(S \vdash p, S' \vdash p') \in R$ iff $p \in \overline{S'}$.*

For a set S of assumptions, we let $\overline{S} = \{\overline{a} \mid a \in S\}$; moreover, we write $asms(E) = \bigcup_{S \vdash p \in E} S$ to denote the set of assumptions of a given set of arguments $E \subseteq A$.

ABA and AFs are closely related (see (Čyras et al. 2018)).

Proposition 2.5. *Given an ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$, its corresponding AF F and a semantics $\sigma \in \{gr, co, pr, stb\}$. If $E \in \sigma(F)$ then $asms(E) \in \sigma(D)$; if $S \in \sigma(D)$ then $\{S' \vdash p \mid \exists S' \subseteq S, R \subseteq \mathcal{R} : S' \vdash_R p\} \in \sigma(F)$.*

3 Dynamics in ABA

In this section, we discuss enforcement and strong equivalence notions for ABA. We show that, in contrast to analogous settings in abstract argumentation, deciding enforceability as well as strong equivalence is intractable.

We consider expansions of ABA frameworks that allow for the addition of rules and assumptions. We consider expansions componentwise, i.e., given $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ and $H = (\mathcal{L}, \mathcal{R}', \mathcal{A}', ^{-})$ with $\mathcal{A} \cap \mathcal{A}' = \emptyset$, we denote the *expansion* of D by H as $D \cup H = (\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A} \cup \mathcal{A}', ^{-})$.

3.1 Enforcement

Let us now introduce our version of the ABA enforcement problem. There are several variants of this problem proposed in the literature, based on credulous vs. skeptical reasoning and positive vs. negative enforcement. In order to keep our presentation concise we focus on one particular problem and demonstrate how our proposal can help tackling it. The following notion of enforcement is stated for a fixed class $\mathcal{C}$ of ABA frameworks. If not stated otherwise, we suppose $\mathcal{C}$ consists of all flat finite ABA frameworks.

Definition 3.1. *Given a class $\mathcal{C}$ of ABA frameworks. For $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-}) \in \mathcal{C}$ and a semantics σ, we say that an assumption $a \in \mathcal{A}$ (atom $p \in \mathcal{L}$) is* enforceable *wrt. σ if there is some expansion $H = (\mathcal{L}, \mathcal{R}', \mathcal{A}', ^{-})$ s.t. $D \cup H \in \mathcal{C}$ and a (p, resp.) is credulously accepted wrt σ (and p does not occur in any head of $\mathcal{R}'$) in $D \cup H$.*

We require that a conclusion p cannot be enforced by simply adding a fact with head p since this would trivialize the problem. Now, let us revisit our motivating example.

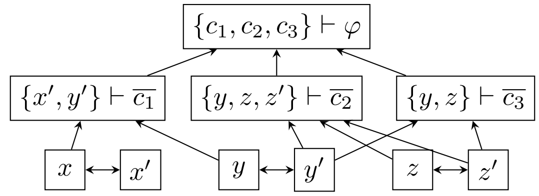


Figure 1: Example of the SAT reduction in Theorem 3.4 for the formula φ given by clauses $\{x, y\}, \{\neg y, z, \neg z\}\{\neg y, \neg z\}$.

Example 3.2. *In our running example ABA D (cf. Example 1.1), we observe that a is credulously enforceable since we can add e.g. the rule "$\overline{b} \leftarrow a$." in order to induce a counterattack from a to b.*

Enforcement is well-studied for abstract argumentation and we would hope that our investigation can benefit from this research. In terms of AFs, our enforcement problem amounts to deciding existence of a suitable expansion; corresponding (im-)possibility results can be found in (Baumann and Brewka 2010). Unfortunately, there are some obstacles which are best explained by our running example D'.

Example 3.3. *Consider the ABA framework D' (cf. Example 1.1). Judging from the associated AF $F_{D'}$, assumption a is easily enforceable by adding some argument attacking x_2. However, since this argument stems from a fact "$\overline{a} \leftarrow$." it is not possible to add rules inducing a counterattack to x_2.*

As this example nicely illustrates, the structure of the ABA framework imposes constraints on the effect an additional argument may have. This observation, unfortunately, renders the corresponding research on enforcement for AFs inapplicable to our situation.

Hence, the next natural question would be whether this issue can be fixed by somehow making the classical enforcement results work for ABA as well. That is, can we perform some simple syntactical check and decide in polynomial time whether some atom is enforceable? As it turns out, under standard assumptions the answer is negative again.

Theorem 3.4. *Deciding whether an assumption a (conclusion p) is enforceable in a given ABA framework D w.r.t. a semantics $\sigma \in \{gr, co, pr, stb\}$ is* NP*-complete.*

Proof (conclusion enforcement, $\sigma = stb$). We reduce SAT: Given a CNF φ with clauses $\mathcal{C}$ over variables in X, we define a corresponding ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ with

- $\mathcal{A} = X \cup \{x' \mid x \in X\} \cup \mathcal{C}$ with $\overline{a} = a$ for all $a \in \mathcal{A}$;
- for each $c \in \mathcal{C}$, $\mathcal{R}$ contains a rule r with $head(r) = c'$ and $body(r) = \{x \mid \neg x \in c\} \cup \{x' \mid x \in c\}$, moreover, $\mathcal{R}$ contains a rule $\varphi \leftarrow \mathcal{C}$.

Instantiating this ABA as AF results in the well-known standard reduction (cf. (Dvorák and Dunne 2018, Reduction 3.6)); an example is given in Figure 1. It can be shown that φ is satisfiable iff φ is inferred by a stable extension of D.

Observe that it is not possible to defend the assumption set $\mathcal{C}$ by adding novel rules to D since this would require to introduce rules with head $\overline{x}$ or $\overline{x'}$. That is, in case φ is not satisfiable, it holds that φ cannot be enforced in D. $\square$

3.2 Strong Equivalence

We consider the following natural notion of strong equivalence, where we assume that the background information and the assumptions are fixed and knowledge bases are characterized by their rules. If the class $\mathcal{C}$ is not specified, we suppose $\mathcal{C}$ contains all flat finite ABA frameworks.

Definition 3.5. *Let $\mathcal{C}$ be a class of ABA frameworks. Two ABA frameworks $D \in \mathcal{C}$ and $D' \in \mathcal{C}$ are* strongly equivalent w.r.t. σ *($D \equiv_s^\sigma D'$) if $\sigma(D \cup H) = \sigma(D' \cup H)$ for each expansion H satisfying $D \cup H \in \mathcal{C}$ as well as $D' \cup H \in \mathcal{C}$.*

We mention an alternative approach that considers equivalence on conclusion-level instead of assumption-level: Two ABAs $D, D' \in \mathcal{C}$ are *strongly conclusion-equivalent* to each other if they agree on their conclusion-extensions under all possible expansions ($D \equiv_s^{Th_\sigma} D'$ for short), formally, $\{Th(E) \mid E \in \sigma(D \cup H)\} = \{Th(E) \mid E \in \sigma(D' \cup H)\}$ for each expansion H s.t. $D \cup H \in \mathcal{C}$ and $D' \cup H \in \mathcal{C}$. It turns out that both notions are equivalent; also, it is equally possible to consider only proper conclusion-extensions.

Proposition 3.6. *For $\sigma \in \{gr, co, pr, stb\}$, for any two ABAs D and D', it holds that $D \equiv_s^{Th_\sigma} D'$ iff $D \equiv_s^\sigma D'$, even when considering only proper conclusion-extensions.*

Moreover, assumption-sets of strongly equivalent ABA frameworks agree on their conclusions.

Proposition 3.7. *For $\sigma \in \{gr, co, pr, stb\}$, for ABAs D, D', if $D \equiv_s^\sigma D'$ then $Th_D(E) = Th_{D'}(E)$ for all $E \in \sigma(D)$.*

As for the enforcement problem, strong equivalence is well-studied for abstract argumentation (Oikarinen and Woltran 2011). There is even another similarity: Deciding strong equivalence is also tractable for abstract argumentation, since one only needs to compute the so-called kernels of both AFs and check their syntactical coincidence.

However, we are facing the same issue regarding ABA.

Example 3.8. *Our running example ABAs D and D' both induce the same AF as we saw before. Hence, the abstract AFs F_D and $F_{D'}$ are clearly strongly equivalent. However, already from the fact that a is enforceable in D but not in D' we can infer that D and D' are not strongly equivalent.*

To complete the picture, we mention that for strong equivalence, we again do not expect the a polynomial-time syntactical check. Adapting the proof of Theorem 3.4 shows that deciding strong equivalence is coNP-complete.

Theorem 3.9. *Deciding whether two ABA frameworks are strongly equivalent w.r.t. a given semantics $\sigma \in \{gr, co, pr, stb\}$ is* coNP*-complete.*

4 An Instantiation For Dynamics

In this section, we present cvAFs ("conclusion and vulnerability augmented AFs") which extend Dung-style AFs with additional information concerning the occurring arguments. Our cvAFs incorporate the observation we made earlier in the introduction: arguments are typically characterized by their *conclusion* and their potential weaknesses (*vulnerabilites*) on which they can be attacked.

4.1 Instantiated Arguments

Recall our motivating Example 1.1. Given the ABA framework D from the introduction, our goal is to preserve conclusions and vulnerabilities during instantiation.

For this we observe that structured argumentation instantiation procedures make use of some meta-level information where each argument is augmented with the two aforementioned features, namely the conclusions as well as the vulnerabilities. However, when instantiating a Dung-style AF this information is lost. We therefore consider a slight generalization of AFs based on the following notion of instantiated arguments, which carry the information we require.

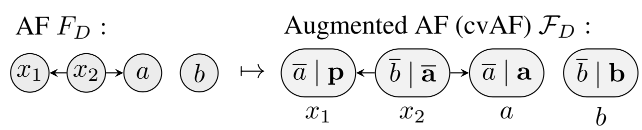
Definition 4.1. *Given a set $\mathcal{L}$ of sentences. An* instantiated argument *is a tuple $x = (vul(x), cl(x))$ where $vul(x) \subseteq \mathcal{L}$ are the* vulnerabilities *and $cl(x) \in \mathcal{L}$ is the* conclusion *of x.*

Instantiated arguments are a flexible tool and may stem from an arbitrary instantiation procedure which makes use of conclusions and vulnerabilities in a certain sense, e.g. ABA, ASPIC, or logic-based argumentation.

With our notion of instantiated arguments at hand we are ready to formally introduce cvAFs as generalization of AFs by replacing abstract arguments with instantiated arguments.

Definition 4.2. *A cvAF is a tuple $\mathcal{F} = (A, R)$ where A is a set of instantiated arguments and $R \subseteq A \times A$.*

An example of a cvAF is given by the representation of our running example as cvAF (cf. $\mathcal{F}_D$ below). Here, each argument contains its vulnerabilities (left) and its conclusion (right, in boldface), e.g., argument x_1 has a single vulnerability $\overline{a}$ and conclusion p.



We consider a crucial property based on the following observation that appears in many structured argumentation formalisms: outgoing attacks usually depend on the conclusion of the attacking argument while incoming attacks are characterized by the vulnerabilities. This means that arguments with conclusion p attack all arguments with vulnerability p. A cvAFs adhering to this property is called well-formed.

Definition 4.3. *A cvAF is called $\mathcal{F} = (A, R)$* well-formed *if $(x, y) \in R$ iff $cl(x) \in vul(y)$ for each $x, y \in A$.*

Let us now turn our attention towards the semantics. By utilizing *instantiated* instead of *abstract* arguments when instantiating a knowledge base $\mathcal{K}$, we find a cvAF $\mathcal{F}_\mathcal{K}$ whose structure is isomorphic to the standard instantiation $F_\mathcal{K}$, but the arguments in $\mathcal{F}_\mathcal{K}$ carry more information. Since we do not intend to alter the meaning of the instantiation procedure, we stick to the classical AF semantics when evaluating $\mathcal{F}_\mathcal{K}$; we mention however that cvAFs are flexible in the sense that we can either focus on the arguments themselves or their conclusions.

Definition 4.4. *Given a cvAF $\mathcal{F}$ and an AF semantics σ. We call $\sigma(\mathcal{F})$ the σ-argument-extensions of $\mathcal{F}$ and $\sigma_{cl}(\mathcal{F}) := \{cl(E) \mid E \in \sigma(F)\}$ the σ-conclusion-extensions of $\mathcal{F}$.*

Note that for well-formed cvAFs, each σ-argument-extension corresponds to a unique σ-conclusion-extension.

Proposition 4.5. *For each well-formed cvAF $\mathcal{F}$, for each $\sigma \in \{gr, co, pr, stb\}$, it holds that $|\sigma(\mathcal{F})| = |\sigma_{cl}(\mathcal{F})|$.*

Let us now see or formalism at work when applied to ABA frameworks. We use the usual procedure, but consider instantiated arguments where the conclusions and underlying assumptions are made explicit. So suppose from our ABA framework we can construct an argument of the form $S \vdash p$. An argument of this form yields an instantiated argument $x = (vul(x), cl(x)) = (\overline{S}, p)$. Therefore, a natural instantiation with instantiated arguments is as follows.

Definition 4.6. *For an ABA framework D, $\mathcal{F}_D = (A, R)$ is the cvAF with instantiated arguments $A = \{(\overline{S}, p) \mid S \vdash p\}$ and $(x, y) \in R$ iff $cl(x) \in vul(y)$.*

Our cvAF instantiation is a faithful generalization of the usual one; moreover, each instantiated cvAF is well-formed.

Proposition 4.7. *For each ABA D, its associated cvAF $\mathcal{F}_D$ is well-formed. Moreover, for each $\sigma \in \{gr, co, pr, stb\}$,*

1. *if $S \in \sigma(D)$ then $\{(\overline{S'}, p) \in A \mid S' \subseteq S\} \in \sigma(\mathcal{F}_D)$;*
2. *if $E \in \sigma(\mathcal{F}_D)$ then $\bigcup_{(\overline{S},p)\in E} S \in \sigma(D)$.*

Furthermore, $\{Th_D(S) \mid S \in \sigma(D)\} = \sigma_{cl}(\mathcal{F}_D)$.

Indeed, the ABA framework D from our running example yields the cvAF $\mathcal{F}_D$ depicted above. Moreover, $\sigma(\mathcal{F}_D) = \{\{x_2, b\}\}$ and hence $\sigma_{cl}(\mathcal{F}_D) = \{\{\bar{b}, \}\}$ correctly reflects the semantics of D. We also mention that when instantiating D' from Example 1.1 we obtain a similar cvAF, but x_2 does not have any vulnerability. Hence we are indeed able to distinguish the two instantiations as desired.

Since our formalism of interest yields well-formed cvAFs, we restrict our studies to *well-formed cvAFs only*.

Assumption 4.8. *In the remaining part of the paper, we assume that each cvAF is well-formed.*

4.2 cvAFs and Dynamics

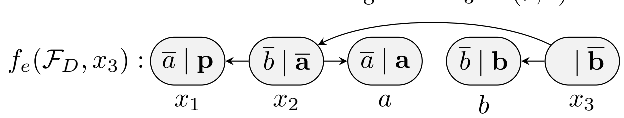
We are ready to investigate dynamics in structured argumentation by means of cvAFs. Suppose we are given a knowledge base $\mathcal{K}$ and the instantiated cvAF $\mathcal{F}_{\mathcal{K}}$. If we want to move to a superset $\mathcal{K} \cup \mathcal{H}$ we can construct $\mathcal{F}_{\mathcal{K}\cup\mathcal{H}}$ immediately by inspecting the relevant conclusions and vulnerabilities. This idea can be formalized as follows.

Definition 4.9. *Given a cvAF $\mathcal{F} = (A, R)$ and an instantiated argument x we define the expansion $f_e(\mathcal{F}, x)$ of $\mathcal{F}$ with x by letting $f_e(\mathcal{F}, x) = (A \cup \{x\}, R_x)$ be the cvAF where*

$$R_x = R \cup \{(x, y) \mid y \in A, cl(x) \in vul(y)\} \cup \{(y, x) \mid y \in A, cl(y) \in vul(x)\}.$$

We stipulate that $f_e(\mathcal{F}, X)$ is a shorthand for successively expanding $\mathcal{F}$ with each $x \in X$ in an arbitrary order.

Example 4.10. *Let D be our running example and $\mathcal{F}_D$ its instantiated cvAF as depicted above. Adding a fact "$\bar{b}$." yields an additional instantiated argument $x_3 = (\emptyset, \bar{b})$:*



4.3 cvAFs and Atomic ABA Frameworks

Our cvAFs are closely related to a certain fragment of ABA in dynamic scenarios which we will discuss next.

Definition 4.11. *Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework. A rule r is* atomic *if $body(r) \subseteq \mathcal{A}$. The ABA framework is* atomic *if each rule $r \in \mathcal{R}$ is atomic.*

There are two decisive observations we make about atomic ABA frameworks. The first is concerning cvAFs.

Lemma 4.12. *Given an atomic ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$.*

- *If $a \in \mathcal{A}$, then $\mathcal{F}_{D\cup\{r\}} = f_e(\mathcal{F}_D, x)$ with $x = (\bar{a}, a)$.*
- *For each atomic (in D) rule $r = p \leftarrow S$, it holds that $\mathcal{F}_{D\cup\{r\}} = f_e(\mathcal{F}_D, x)$ with $x = (\overline{S}, p)$.*
- *If $x = (\overline{S}, p)$, then $\mathcal{F}_{D\cup H} = f_e(\mathcal{F}_D, x)$ with $H = (\mathcal{L}, \mathcal{R}', \mathcal{A}', ^-)$ with $\mathcal{R}' = \{(p \leftarrow S)\}$ and $\mathcal{A}' = \overline{S}$.*

Second, by moving from general to atomic ABA frameworks we do not lose expressive power; each framework can be transformed into an atomic one.

Proposition 4.13. *For each ABA D there is an atomic ABA D' such that $\sigma(D) = \sigma(D')$ for each considered semantics.*

We mention however that this transformation might result in an exponential blow-up in the number of rules. However, given an atomic ABA framework D we can be sure that the instantiated cvAF $\mathcal{F}_D$ is of linear size in D.

Proposition 4.14. *If $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ is atomic, then the cvAF $\mathcal{F}_D$ consists of $|\mathcal{R}| + |\mathcal{A}|$ arguments.*

5 The cvAF Enforcement Problem

In this section we develop a notion of the enforcement problem for cvAFs and establish criteria for deciding enforcability. At first glance, this yields results applicable to atomic ABAs due to Lemma 4.12; we will however discuss some subtle details of the notions which one needs to be aware of.

In line with our enforcement notion from Definition 3.1, we define conclusion enforcement for cvAFs by requiring that no new argument with the target conclusion is introduced. In addition, we introduce a natural notion of argument enforcement.

Definition 5.1. *Let $\mathcal{F} = (A, R)$ be a cvAF and σ a semantics. We say that a conclusion p is σ-*enforceable *if there is a set X of instantiated arguments s.t. $cl(x) \neq p$ for each $x \in X$ and p is credulously accepted in $f_e(\mathcal{F}, X)$. An argument $x \in A$ is σ-*enforceable *if there is a set X of instantiated arguments s.t. x is credulously accepted in $f_e(\mathcal{F}, X)$.*

Example 5.2. *Let $\mathcal{F}_D$ be our running example cvAF and consider the expansion $f_e(\mathcal{F}_D, x_3)$ with $x_3 = (\emptyset, \bar{b})$ (cf. Example 4.10). Since $co(f_e(\mathcal{F}_D, x)) = \{\{a, x_1, x_3\}\}$ with $cl(x_1) = p$ we obtain that conclusion p is co-enforceable.*

In the following we establish criteria to decide whether arguments and conclusions are enforceable in cvAFs. It turns out that it suffices to focus on argument enforcement:

Proposition 5.3. *Let $\mathcal{F} = (A, R)$ be a cvAF and σ a semantics. A conclusion $c \in cl(A)$ is enforceable iff there is some $x \in A$ with $cl(x) = c$ s.t. x is enforceable.*

The possible modifications of a cvAF are determined by the conclusions and vulnerabilities of its arguments. It is thus not possible to consider arbitrary expansions. We already saw this for our running example $\mathcal{F}_{D'}$ where a is not enforceable since it is attacked by some argument without any vulnerability (cf. Example 3.3).

In general, arguments of this kind will always be defeated in complete-based semantics. This is not only the case within the given cvAF, but also for any conceivable expansion. Motivated by this observation, we call such arguments *strongly defeated*.

Definition 5.4. *For a cvAF $\mathcal{F} = (A, R)$, $x \in A$ is* strongly defeated *if there is $y \in A$ with $(y, x) \in R$ and $vul(y) = \emptyset$.*

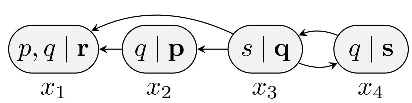
Example 5.5. *In our running example cvAF stemming from instantiating D', the argument x_1 is strongly defeated. In fact, it is verifiable with reasonable effort that x_2 is part of the grounded extension in any possible expansion $f_e(\mathcal{F}, X)$.*

The following proposition formalizes that the behavior we observed in the previous example generalizes to any cvAF and thus our intuition about strong defeat is indeed correct.

Proposition 5.6. *Let $\mathcal{F} = (A, R)$ be a cvAF. If $x \in A$ is strongly defeated, then for each set X of instantiated arguments, the grounded extension of $f_e(\mathcal{F}, X)$ attacks x.*

Consequently, we infer that strongly defeated arguments can never be enforced. It is therefore a reasonable conjecture that an argument is enforceable iff it is not strongly defeated. However, as the following example illustrates, the notion of strong defeat is not yet general enough.

Example 5.7. *Consider the cvAF $\mathcal{F}$ depicted below.*

$\mathcal{F}$:


Suppose we want to enforce x_1. In order to achieve this goal we have to add an argument defeating x_2. However, the only vulnerability of x_2 is q and due to $q \in vul(x_1)$, such an argument would defeat x_1 as well.

In general, if there is some argument y with $(y, x) \in R$ and $vul(y) \subseteq vul(x)$, then x can never be defended by a conflict-free set. We call arguments of this kind *strongly unacceptable* since this holds also true for any expansion.

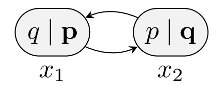
Definition 5.8. *For a cvAF $\mathcal{F} = (A, R)$, $x \in A$ is* strongly unacceptable *if there is $y \in A$ with $(y, x) \in R$ and $vul(y) \subseteq vul(x)$.*

By definition, each strongly defeated argument is strongly unacceptable. For $\sigma \in \{co, pr, stb\}$ we are now ready to state our enforcement results.

Theorem 5.9. *Let $\mathcal{F} = (A, R)$ be a cvAF and suppose $\sigma \in \{co, pr, stb\}$. An argument $x \in A$ is σ-enforceable if and only if it is not strongly unacceptable.*

For grounded semantics, however, we need to consider further unacceptability notions. The reason why Theorem 5.9 does not hold for grounded semantics is that an argument might be capable of defending itself, but is still not part of the iterative procedure which yields the grounded extension. To illustrate this we consider the following example.

Example 5.10. *Suppose we aim to gr-enforce x_1 in $\mathcal{F}$:*

$\mathcal{F}$:


Since $gr(\mathcal{F}) = \emptyset$ we would have to introduce an argument defeating x_2 in order defend x_1 from the in-coming attack. However, an argument achieving this would possess p as conclusion which we want to avoid for this version of the enforcement notion. Indeed, x_1 is not gr-enforceable.

In general, for grounded semantics we require a notion which is similar to strong unacceptability, while taking the special case we just illustrated into account.

Definition 5.11. *For a cvAF $\mathcal{F} = (A, R)$, $x \in A$ is* strongly gr-unacceptable *if there is $y \in A$ with $(y, x) \in R$ and $vul(y) \setminus \{cl(x)\} \subseteq vul(x)$.*

The following condition characterizes gr-enforceability for cvAFs. Although it may appear technical at first glance, it simply ensures that an argument z can be defeated without attacking x, y, or introducing the target claim $cl(x) = cl(y)$.

Proposition 5.12. *Let $\mathcal{F} = (A, R)$ be a cvAF. An argument $x \in A$ is gr-enforceable if and only if one of the following two conditions hold:*

- *x is not strongly gr-unacceptable,*
- *there is some $y \in A$ with $cl(y) = cl(x) = q$ s.t.*
  - *if z attacks y, then $vul(z) \setminus (vul(x) \cup vul(y) \cup \{q\}) \neq \emptyset$,*
  - *if z attacks x, then $q \in vul(z)$ or $vul(z) \setminus (vul(x) \cup vul(y)) \neq \emptyset$.*

Let us now discuss corresponding results for conclusion enforcement. To enforce a conclusion $p \in cl(A)$ we need to enforce an argument $x \in A$ with $cl(x) = p$. Thus, as a corollary of Theorem 5.9 and Proposition 5.12 we obtain:

Corollary 5.13. *Let $\mathcal{F} = (A, R)$ be a cvAF and $\sigma \in \{ad, co, pr, stb\}$. An argument $a \in A$ is σ-enforceable if and only if it is not strongly unacceptable; it is gr-enforceable if and only if it is not strongly gr-unacceptable.*

Consequences for ABA. The introduced unacceptability notions yield syntactical conditions to decide the cvAF enforcement problem in polynomial time. In view of this, it might seem that Lemma 4.12 now implies tractability of the enforcement problem for atomic ABA frameworks. However, when inspecting the construction for the proof of Theorem 3.4 we see that the constructed ABA framework is atomic itself.

Corollary 5.14. *Deciding whether a (p) is enforceable wrt σ is* NP*-complete even for atomic ABA frameworks.*

The reason why this is no contradiction to tractability in cvAFs is rather subtle: When considering an arbitrary expansion $f_e(\mathcal{F}, X)$, it might happen that the resulting cvAF does not correspond to a flat ABA framework anymore due to $cl(X) \cap \mathcal{A} \neq \emptyset$. So even if we start with a cvAF corresponding to some flat ABA framework, we do not have a one to one correspondence between expansions of the cvAF and flat ABA framework extending the initial one.

To ensure that it is not necessary to introduce arguments with assumptions as conclusion when enforcing an argument $x \in A$, we consider a particular fragment of ABA where assumptions do not have out-going attacks.

Definition 5.15. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework. We say D has* separated contraries *if $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$.*

Indeed, we are now ready to introduce a tractable fragment for the ABA enforcement problem.

Theorem 5.16. *Deciding whether an argument or conclusion is enforceable for atomic flat ABAs with separated contraries is tractable.*

We want to emphasize that moving from flat ABA to flat atomic ABA does not change the complexity class of the enforcement problem; but additional requiring separated contraries does, i.e., we found a rather minor condition pushing the enforcement problem over the edge to tractability.

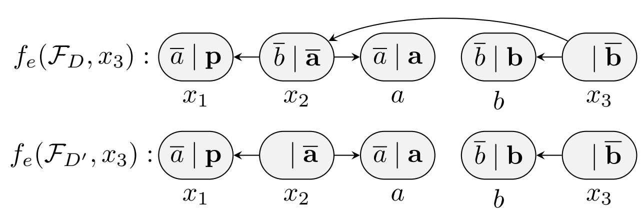
6 The cvAF Strong Equivalence Problem

In this section, we establish methods to decide strong equivalence for cvAFs. We define further unacceptability notions, tailored for this setting. In accordance with the standard literature on strong equivalence we then can decide this problem for two cvAFs by comparing their so-called *kernels*, that is, we transform both cvAFs into a semantics-dependent normal form. Let us point out the following crucial difference: In contrast to strong equivalence characterizations in Dung AFs where kernels are constructed by removing redundant *attacks*, we identify redundant *arguments*. The kernels in cvAFs are constructed by *removing* as well as *manipulating* arguments that fall in certain redundancy categories.

We start by defining an appropriate strong equivalence notion for cvAFs.

Definition 6.1. *Two cvAFs $\mathcal{F}$, $\mathcal{G}$ are* strongly equivalent *wrt a semantics σ, for short $\mathcal{F} \equiv^{\sigma}_{s} \mathcal{G}$, if for each set X of instantiated arguments $\sigma_{cl}(f_e(\mathcal{F}, X)) = \sigma_{cl}(f_e(\mathcal{G}, X))$ holds.*

Example 6.2. *Consider again the cvAFs $\mathcal{F}_D$ and $\mathcal{F}_{D'}$ from Example 1.1. Judging from earlier results we anticipate that they are not strongly equivalent to each other. Indeed, adding the argument $x_3 = (\emptyset, \overline{b})$ yields the following cvAFs:*



Now, $\{a, p, \overline{b}\}$ is stable in $f_e(\mathcal{F}_D, x_3)$ but not in $f_e(\mathcal{F}_{D'}, x_3)$. We obtain that $\mathcal{F}_D$ and $\mathcal{F}_{D'}$ are not strongly equivalent wrt stable semantics, i.e., $\mathcal{F}_D \not\equiv^{stb}_{s} \mathcal{F}_{D'}$.

In the above example, it was quite easy to come up with an appropriate counter example. Not only that finding a counter example might be more involved in other situations, it is usually not possible to verify strong equivalence by testing all possible expansions because there might be infinitely many of them. Instead, we identify, for each semantics, a specific kernel – checking strong equivalence then reduces to computing and comparing the respective kernels.

Let us first reconsider the unacceptability notions from Section 5. We have shown that strongly defeated arguments cannot be enforced; in fact, they can be removed without changing the σ_{cl}-extensions.

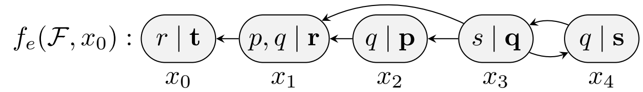
Proposition 6.3. *Given a cvAF $\mathcal{F} = (A, R)$, a semantics $\sigma \in \{gr, co, pr, stb\}$ and a strongly defeated argument $x \in A$. Then $\sigma_{cl}(\mathcal{F}) = \sigma_{cl}(\mathcal{F} \setminus \{x\})$.*

Likewise, for stable semantics, strongly unacceptable arguments can be deleted without affecting the outcome.

Proposition 6.4. *For a cvAF $\mathcal{F} = (A, R)$ and a strongly unacceptable argument $x \in A$, $stb_{cl}(\mathcal{F}) = stb_{cl}(\mathcal{F} \setminus \{x\})$.*

Considering grounded, complete, and preferred semantics, we observe that strongly unacceptable arguments are not necessarily defeated – removing them thus potentially results in a change of the σ_{cl}-extensions.

Example 6.5. *Consider the cvAF $\mathcal{F}$ from Example 5.7 together with a new argument $x_0 = (\{r\}, t)$:*



$f_e(\mathcal{F}, x_0)$ has three complete conclusion-extensions: $\emptyset$ (the grounded extension), $\{s, p, t\}$, and $\{q, t\}$. Recall that x_1 is strongly unacceptable w.r.t. x_2. Removing x_1 would make x_0 unattacked, changing the grounded extension to $\{t\}$.

Strongly unacceptable arguments can neither be enforced nor deleted in such situations. This means that on semantics level, it is not possible to distinguish if such arguments are self-attacking or not (analogous observations hold for gr-unacceptable arguments wrt grounded semantics).

Proposition 6.6. *Given a cvAF $\mathcal{F} = (A, R)$, a semantics $\sigma \in \{gr, co, pr, stb\}$ and a strongly unacceptable argument $x \in A$ and let $x' = (vul(x) \cup \{cl(x)\}, cl(x))$. Then $\sigma(\mathcal{F}) = \sigma(f_e(\mathcal{F} \setminus \{x\}, x'))$. If x is strongly gr-unacceptable, then $gr(\mathcal{F}) = gr((f_e(\mathcal{F} \setminus \{x\}, x'))$.*

Next we consider a redundancy notion that also appears in different contexts (Dvorák and Woltran 2020): We call an argument x *redundant* if there is some other argument y having the same conclusion but possesses less vulnerabilities.

Definition 6.7. *For a cvAF $\mathcal{F} = (A, R)$, $x \in A$ is* redundant *if there is $y \in A$ with $cl(y) = cl(x)$ and $vul(y) \subsetneq vul(x)$.*

Example 6.8. *The argument x_2 from the cvAF $\mathcal{F}_D$ from our running example is redundant wrt $x = (\emptyset, \overline{a})$ because $cl(x) = cl(x_2) = \overline{a}$ and $vul(x) = \emptyset \subsetneq \{\overline{b}\} = vul(x_2)$.*

Redundant arguments can be removed without changing the conclusion-based extensions of a given cvAF.

Proposition 6.9. *For a cvAF $\mathcal{F} = (A, R)$, a semantics $\sigma \in \{gr, co, pr, stb\}$ and a redundant argument $x \in A$, it holds that $\sigma_{cl}(\mathcal{F}) = \sigma_{cl}(\mathcal{F} \setminus \{x\})$.*

We are ready to define the complete and the grounded kernel for cvAFs. Following Proposition 6.6, the first adjustment we carry out is a modification on vulnerability level:

Each unacceptable argument x is turned into a self-attacker by adding $cl(x)$ to $vul(x)$; for grounded semantics, we modify each gr-unacceptable argument instead. In the next step, we remove all strongly defeated and redundant arguments.

Definition 6.10. *For a cvAF $\mathcal{F} = (A, R)$, let X denote the set of all strongly unacceptable arguments in A and let*

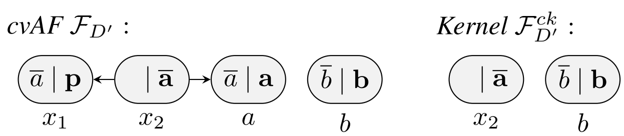
$$(A', R') = f_e(\mathcal{F}\setminus X, \{(vul(x)\cup\{cl(x)\}, cl(x)) \mid x \in X\}).$$

We define the complete kernel $\mathcal{F}^{ck} = (A^{ck}, R^{ck})$ *with*

$$A^{ck} = A' \setminus \{x \in A' \mid x \text{ is str. defeated or redundant}\},$$

and $R^{ck} = R' \cap (A^{ck} \times A^{ck})$. The grounded kernel $\mathcal{F}^{gk}$ *is defined analogously by replacing X with the set of all strongly gr-unacceptable arguments $x \in A$.*

Example 6.11. *The cvAF $\mathcal{F}_D$ from our running example coincides with its complete and grounded kernel since no arguments are strongly defeated, unacceptable or redundant. That is, we obtain $\mathcal{F}_D^{ck} = \mathcal{F}_D^{gk} = \mathcal{F}_D$. Recalling that x_2 is strongly unacceptable in $\mathcal{F}_{D'}$ we obtain the following picture (observe that $\mathcal{F}_{D'}^{ck} = \mathcal{F}_{D'}^{gk}$ also holds in this case):*



Next we consider a special case of strong unacceptability that affects only preferred semantics.

Definition 6.12. *For a cvAF $\mathcal{F} = (A, R)$, $x \in A$ is* strongly pr-unacceptable *if x is strongly unacceptable w.r.t. $y \in A$ and $vul(y) = \{cl(x)\}$.*

Observe that each strongly pr-unacceptable argument is self-attacking because $vul(y) = \{cl(x)\} \subseteq vul(x)$. It turns out that such arguments can be removed without changing the conclusion-based preferred extensions.

Proposition 6.13. *For a cvAF $\mathcal{F} = (A, R)$ and a strongly pr-unacceptable argument $x \in A$, $pr_{cl}(\mathcal{F}) = pr_{cl}(\mathcal{F}\setminus\{x\})$.*

The preferred kernel refines the complete kernel:

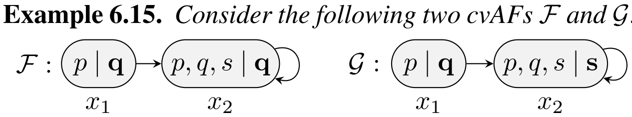
Definition 6.14. *For a cvAF $\mathcal{F} = (A, R)$, let $\mathcal{F}^{ck} = (A^{ck}, R^{ck})$ be as in Definition 6.10. We define the* preferred kernel $\mathcal{F}^{pk} = (A^{pk}, R^{pk})$ *with*

$$A^{pk} = A^{ck} \setminus \{x \in A^{ck} \mid x \text{ is str. } pr\text{-unacceptable}\},$$

and $R^{pk} = R^{ck} \cap (A^{pk} \times A^{pk})$.

Finally, we consider stable semantics. We start with the crucial observation that the particular conclusion of self-attacking arguments is not of importance.

Example 6.15. *Consider the following two cvAFs $\mathcal{F}$ and $\mathcal{G}$:*



Essentially, $\mathcal{F}$ and $\mathcal{G}$ differ only in the conclusion of the self-attacking argument x_2. Observe that $\mathcal{F}$ and $\mathcal{G}$ both admit the same unique stable extension $\{q\}$. As we will see, this is not a coincidence: For stable semantics, self-attacking arguments are indistinguishable with respect to their conclusion.

Proposition 6.16. *Given a cvAF $\mathcal{F} = (A, R)$ and a self-attacking argument $x \in A$. For any $s \in vul(x)$, it holds that $stb_{cl}(\mathcal{F}) = stb_{cl}(f_e(\mathcal{F}, \{(vul(x), s)\}))$.*

We construct the stable kernel by (i) adding all such missing self-attackers before (ii) removing all redundant, strongly defeated, and strongly unacceptable arguments.

Definition 6.17. *For a cvAF $\mathcal{F} = (A, R)$, let X denote the set of all self-attacking arguments in A and let*

$$(A', R') = f_e(\mathcal{F}, \{(vul(x), s) \mid x \in X, s \in vul(x)\}).$$

We define the stable kernel $\mathcal{F}^{sk} = (A^{sk}, R^{sk})$ *with*

$$A^{sk} = A' \setminus \{x \in A' \mid x \text{ is str. defeated, str. unacceptable, or redundant}\},$$

and $R^{sk} = R' \cap (A^{sk} \times A^{sk})$.

Iterative application of the previously established propositions shows that the kernels are semantics-preserving. Below we write $\mathcal{F}^{k(\sigma)}$ to denote the σ-kernel of $\mathcal{F}$.

Lemma 6.18. $\sigma(\mathcal{F}) = \sigma(\mathcal{F}^{k(\sigma)})$ *for every cvAF $\mathcal{F}$ and semantics σ considered in this paper.*

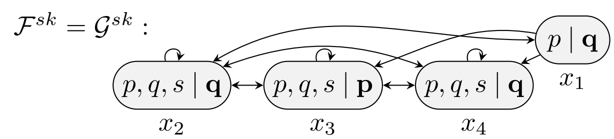
The syntactical coherence is guaranteed by the following technical lemma which states that the order of adding additional arguments is not of importance.

Lemma 6.19. *For every cvAF $\mathcal{F}$, for every instantiated argument x, $(f_e(\mathcal{F}, \{x\}))^{k(\sigma)} = (f_e(\mathcal{F}^{k(\sigma)}, \{x\}))^{k(\sigma)}$.*

We are ready to present our characterization result for cvAF strong equivalence: strong equivalence can be decided by computing and comparing the respective kernels.

Theorem 6.20. *For any two cvAFs $\mathcal{F}$ and $\mathcal{G}$, for any $\sigma \in \{gr, co, pr, stb\}$, $\mathcal{F} \equiv_s^\sigma \mathcal{G}$ if and only if $\mathcal{F}^{k(\sigma)} = \mathcal{G}^{k(\sigma)}$.*

Recall that the cvAFs $\mathcal{F}_D$ and $\mathcal{F}_{D'}$ from our running example are not strongly equivalent to each other as already observed in Example 6.2. Theorem 6.20 now gives a criteria to check strong equivalence for any two cvAFs without searching for counter-examples. Considering for instance the cvAFs $\mathcal{F}$ and $\mathcal{G}$ from Example 6.15, we obtain



By Theorem 6.20 we thus can conclude that $\mathcal{F}$ and $\mathcal{G}$ are strongly equivalent w.r.t. stable semantics without testing each possible expansion.

Consequences for ABA By transferring the above results in the context of ABA we obtain that deciding strong equivalence for flat, atomic ABA frameworks with separated contraries is tractable.

Theorem 6.21. *For two atomic, flat ABA frameworks D, D' with separated contraries, deciding $D \equiv_s^\sigma D'$ is tractable.*

As in the case of enforcement, we want to emphasize that moving from flat ABA to flat atomic ABA does not change the complexity class of this problem. However, if we additionally require that the frameworks have separated assumptions we obtain the desired tractable fragment.

7 Consequences for Logic Programs

The attentive reader may have observed that we inferred our tractability results by changing the instantiation from AFs to cvAFs and then applying results for cvAFs instead of directly investigating ABA frameworks as deductive systems. While this technique seems cumbersome as a first glance, the established results for cvAFs turn out to be a convenient tool which we can now apply to logic programs almost immediately. This section demonstrates this approach.

We consider the logic programming semantics discussed in (Caminada et al. 2015b); for those, a close correspondence to AF semantics has been shown. We assume the reader to be familiar with these concepts; an introduction to LPs can be found in the supplementary material. We define credulous acceptance of an atom a occurring in P in the expected way. Given an LP P, the corresponding instantiated cvAF is denoted by $\mathcal{F}_P$. We proceed as for ABA frameworks by applying the cvAF results. Since they have been shown already, we only have left to give an LP version of Proposition 4.7 as well as Lemma 4.12.

Proposition 7.1. *For each LP P and its associated cvAF $\mathcal{F}_P$, for each $\sigma \in \{gr, co, pr, stb\}$, it holds that*

- *if $X \in \sigma(P)$ then $\{(X', s) \in A \mid X' \subseteq X\} \in \sigma(\mathcal{F}_P)$;*
- *if $E \in \sigma(\mathcal{F}_P)$ then $\bigcup_{(X,s)\in E} X \in \sigma(D)$.*

Lemma 7.2. *Given an atomic LP P and $B = \{b_1, \ldots, b_n\}$.*

- *For each atomic rule $r = c \leftarrow not\ b_1, \ldots, not\ b_n$, it holds that $\mathcal{F}_{P\cup\{r\}} = f_e(\mathcal{F}_P, x)$ with $x = (B, c)$.*
- *For each argument $x = (B, c)$, it holds that $\mathcal{F}_{P\cup\{r\}} = f_e(\mathcal{F}_P, x)$ with $r = c \leftarrow not\ b_1, \ldots, not\ b_n$.*

This suffices in order to efficiently investigate our two problems we considered before. The relation is even much closer since we do not need to handle additional assumptions. We define the LP enforcement problem as follows.

Definition 7.3. *Let P be a logic program and σ a semantics. An atom a is σ-*enforceable *if there is a set R of rules s.t. a is credulously accepted in $P \cup R$ w.r.t. σ.*

The NP-hardness proof we gave for ABA in Theorem 3.4 works analogously for LPs and thus, the enforcement problem is intractable for LPs in general. However, our cvAF results yield tractability for atomic LPs.

Theorem 7.4. *For atomic LPs, deciding whether some atom is enforceable is tractable.*

Analogously we discuss strong equivalence for LPs. As in the corresponding notion of ABA frameworks, we insist that the expansions are atomic as well.

Definition 7.5. *Two atomic LPs P, P' are* atomic strongly equivalent *w.r.t. a semantics σ, for short $P \equiv^{\sigma}_{ase} P'$, if for each atomic LP H $\sigma(P \cup H) = \sigma(P' \cup H)$ holds.*

Without the requirement of P, P', and H being atomic, intractability of strong equivalence is well-known from LP research (Lifschitz, Pearce, and Valverde 2001). Due to our cvAF results, we obtain a tractable fragment here as well.

Theorem 7.6. *Deciding whether two atomic LPs P and Q are atomic strongly equivalent is tractable.*

8 Discussion

We investigated strong equivalence and enforcement for ABA. We showed that in general, both tasks are intractable. Inspired by tractability of the corresponding problems for AFs, we proposed an adjusted instantiation procedure via cvAFs to obtain a closer relation between the knowledge base and the corresponding AF. Due to the close correspondence between atomic ABA frameworks and cvAFs, we demonstrated how our cvAF tractability results yield tractable fragments of ABA for these problems. Finally, we applied our techniques to LPs as well.

Our work extends research on dynamics in argumentation (Rotstein et al. 2010; Rotstein et al. 2008; Snaith and Reed 2017). In the context of AFs, both enforcement and strong equivalence are well-studied (Baumann and Brewka 2010; Oikarinen and Woltran 2011; Baumann 2012a; Wallner, Niskanen, and Järvisalo 2017). Similar to our setting, (Wallner 2020) considers situations where an AF undergoes certain changes, but the permitted modifications are constrained. Constraints on the possibly reachable expansions of a given cvAF are intrinsic to our approach. In a recent paper (Borg and Bex 2021) the authors study under which conditions in a structured argumentation formalism a given formula can be enforced; moreover, strong equivalence is similar in spirit to *stability* (Testerink, Odekerken, and Bex 2019). The authors in (Moguillansky et al. 2008) consider argumentative revision operators in the context of defeasible logic programming. In contrast to our enforcement approach, they allow for the deletion of rules (following some minimal change principle) to warrant a desired conclusion.

Our cvAFs are similar in spirit to CAFs (Dvorák and Woltran 2020) where arguments are equipped with claims. CAFs are well-studied (Dvorák, Rapberger, and Woltran 2020; Dvorák et al. 2021), including research on strong equivalence (Baumann, Rapberger, and Ulbricht 2021). However, since vulnerability awareness is crucial for dynamics in ABA, CAFs would not be suitable in this setting.

The redundancy notions we discussed are similar in spirit to the line of research on syntactic transformations for LPs, see., e.g., (Brass and Dix 1997; Eiter et al. 2004; Wang and Zhou 2005; Lin and Chen 2007), that gave rise to alternative characterizations of strong equivalence (Osorio, Pérez, and Arrazola 2001; Cabalar 2002) and set the ground for further complexity analysis of LP fragments (Eiter et al. 2007).

Future work directions include exploring further formalisms where cvAFs are applicable, i.e., investigating suitability for e.g. logic-based argumentation (Besnard and Hunter 2001). As demonstrated in our LP section, this technique may lead to quickly obtained results. Similarly, finding more reasoning tasks where cvAFs are applicable might contribute to this line of research. It would also be interesting to see under which conditions the requirement of atomic frameworks can be dropped. As a further future research direction we identify the design of efficient algorithms since our tractability results serve as a promising starting point for such an endeavor. We also want to mention that formalisms which incorporate preferences, e.g., ABA+ (Cyras and Toni 2016), do not always yield well-formed cvAFs, so research on this more general case is also worth the effort.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig and by the Austrian Science Fund (FWF) through project W1255-N23 and by the Vienna Science and Technology Fund (WWTF) through project ICT19-065.

References

Baumann, R., and Brewka, G. 2010. Expanding argumentation frameworks: Enforcing and monotonicity results. In *Proc. COMMA'10*, volume 216 of *FAIA*, 75–86. IOS Press.

Baumann, R.; Doutre, S.; Mailly, J.-G.; and Wallner, J. P. 2021. Enforcement in formal argumentation. *J. Appl. Logic* 2:1623–1677.

Baumann, R.; Rapberger, A.; and Ulbricht, M. 2021. Equivalence in argumentation frameworks with a claim-centric view – classical results with novel ingredients. In *Proceedings of the 19th International Workshop on Non-Monotonic Reasoning (NMR'21)*, 71–80. To appear in Proc. AAAI'22.

Baumann, R. 2012a. Normal and strong expansion equivalence for argumentation frameworks. *Artificial Intelligence* 193:18–44.

Baumann, R. 2012b. What does it take to enforce an argument? minimal change in abstract argumentation. In *Proc. ECAI'12*, volume 242 of *FAIA*, 127–132. IOS Press.

Besnard, P., and Hunter, A. 2001. A logic-based theory of deductive arguments. *Artif. Intell.* 128(1-2):203–235.

Bondarenko, A.; Toni, F.; and Kowalski, R. A. 1993. An assumption-based framework for non-monotonic reasoning. In Pereira, L. M., and Nerode, A., eds., *Logic Programming and Non-monotonic Reasoning, Proceedings of the Second International Workshop, Lisbon, Portugal, June 1993*, 171–189. MIT Press.

Borg, A., and Bex, F. 2021. Enforcing sets of formulas in structured argumentation. In Bienvenu, M.; Lakemeyer, G.; and Erdem, E., eds., *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021*, 130–140.

Brass, S., and Dix, J. 1997. Characterizations of the disjunctive stable semantics by partial evaluation. *J. Log. Program.* 32(3):207–228.

Cabalar, P. 2002. A three-valued characterization for strong equivalence of logic programs. In Dechter, R.; Kearns, M. J.; and Sutton, R. S., eds., *Proc. AAAI'02*, 106–111. AAAI Press / The MIT Press.

Caminada, M.; Sá, S.; Alcântara, J.; and Dvořák, W. 2015a. On the difference between assumption-based argumentation and abstract argumentation. *IfCoLog Journal of Logic and its Applications* 2(1):15–34.

Caminada, M.; Sá, S.; Alcântara, J. F. L.; and Dvorák, W. 2015b. On the equivalence between logic programming semantics and argumentation semantics. *Int. J. Approx. Reason.* 58:87–111.

Cyras, K., and Toni, F. 2016. ABA+: assumption-based argumentation with preferences. In Baral, C.; Delgrande, J. P.; and Wolter, F., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, 553–556. AAAI Press.

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.

Doutre, S., and Mailly, J. 2018. Constraints and changes: A survey of abstract argumentation dynamics. *Argument Comput.* 9(3):223–248.

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):321–357.

Dvorák, W., and Dunne, P. E. 2018. Computational problems in formal argumentation and their complexity. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications.

Dvorák, W., and Woltran, S. 2020. Complexity of abstract argumentation under a claim-centric view. *Artif. Intell.* 285:103290.

Dvorák, W.; Greßler, A.; Rapberger, A.; and Woltran, S. 2021. The complexity landscape of claim-augmented argumentation frameworks. In *Proc. AAAI'21*, 6296–6303. AAAI Press.

Dvorák, W.; Rapberger, A.; and Woltran, S. 2020. Argumentation semantics under a claim-centric view: Properties, expressiveness and relation to setafs. In *Proc. KR'20*, 341–350.

Eiter, T.; Fink, M.; Tompits, H.; and Woltran, S. 2004. Simplifying logic programs under uniform and strong equivalence. In Lifschitz, V., and Niemelä, I., eds., *Proc. LPNMR'04*, volume 2923 of *LNCS*, 87–99. Springer.

Eiter, T.; Fink, M.; Tompits, H.; and Woltran, S. 2007. Complexity results for checking equivalence of stratified logic programs. In *Proc. IJCAI'07*, 330–335.

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

Gorogiannis, N., and Hunter, A. 2011. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artif. Intell.* 175(9-10):1479–1497.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL)* 2(4):526–541.

Lin, F., and Chen, Y. 2007. Discovering classes of strongly equivalent logic programs. *J. Artif. Intell. Res.* 28:431–451.

Modgil, S., and Prakken, H. 2017. Abstract rule-based argumentation. *FLAP* 4(8).

Moguillansky, M. O.; Rotstein, N. D.; Falappa, M. A.; García, A. J.; and Simari, G. R. 2008. Argument theory change applied to defeasible logic programming. In Fox, D., and Gomes, C. P., eds., *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, 132–137. AAAI Press.

Oikarinen, E., and Woltran, S. 2011. Characterizing strong equivalence for argumentation frameworks. *Artif. Intell.* 175(14-15):1985–2009.

Osorio, M.; Pérez, J. A. N.; and Arrazola, J. 2001. Equivalence in answer set programming. In Pettorossi, A., ed., *Proc. LOPSTR'01*, volume 2372 of *LNCS*, 57–75. Springer.

Rotstein, N. D.; Moguillansky, M. O.; Falappa, M. A.; García, A. J.; and Simari, G. R. 2008. Argument theory change: Revision upon warrant. In *Proc. COMMA'08*, volume 172 of *FAIA*, 336–347. IOS Press.

Rotstein, N. D.; Moguillansky, M. O.; García, A. J.; and Simari, G. R. 2010. A dynamic argumentation framework. In *Proc. COMMA'10*, volume 216 of *FAIA*, 427–438. IOS Press.

Snaith, M., and Reed, C. 2017. Argument revision. *J. Log. Comput.* 27(7):2089–2134.

Testerink, B.; Odekerken, D.; and Bex, F. 2019. A method for efficient argument-based inquiry. In *International Conference on Flexible Query Answering Systems*, 114–125. Springer.

Wallner, J. P.; Niskanen, A.; and Järvisalo, M. 2017. Complexity results and algorithms for extension enforcement in abstract argumentation. *J. Artif. Intell. Res.* 60:1–40.

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

Wang, K., and Zhou, L. 2005. Comparisons and computation of well-founded semantics for disjunctive logic programs. *ACM Trans. Comput. Log.* 6(2):295–327.