Journal of Artificial Intelligence Research 77 (2023) 563-643 Submitted 12/2022; published 06/2023

On Dynamics in Structured Argumentation Formalisms

Anna Rapberger ANNA.RAPBERGER@TUWIEN.AC.AT
Institute of Logic and Computation,
TU Wien, Austria

Markus Ulbricht MULBRICHT@INFORMATIK.UNI-LEIPZIG.DE
Department of Computer Science, ScaDS.AI,
Leipzig University, Germany

Abstract

This paper is a contribution to the research on dynamics in assumption-based argumentation (ABA). We 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. Notably, 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 abstract framework. 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 dynamics in ABA. This yields the desired tractable fragment. We furthermore discuss consequences for the corresponding problems for logic programs.

1. Introduction

Computational models of argumentation in Artificial Intelligence (AI) (Bench-Capon & Dunne, 2007) establish theoretical foundations to automatize argumentative reasoning. Needless to say, such approaches possess a variety of applications in e.g. legal reasoning, medical sciences, and e-governmental issues (Atkinson, Baroni, Giacomin, Hunter, Prakken, Reed, Simari, Thimm, & Villata, 2017) to mention a few. Arguably the most important booster for this research area was Dung's seminal paper (Dung, 1995) where he proposed *abstract argumentation frameworks* (AFs). In Dung-style AFs arguments are viewed as atomic entities. Dung abstracts away the internal structure of the arguments, i.e., the premises as well as rules required to derive the arguments, and also their conclusions. Consequently, conflicts between arguments are viewed as a mere binary relation and thus Dung obtains a representation of AFs as directed graphs, with the intended meaning of nodes being arguments and edges the attacks between them. AFs have been thoroughly investigated within the last decades (Baroni, Gabbay, Giacomin, & van der Torre, 2018) and therefore provide a solid formal groundwork for argumentative reasoning approaches.

A currently highly relevant area of research in knowledge representation and reasoning is the investigation of dynamic environments, i.e., knowledge bases that change over time

(Gabbay, Giacomin, Simari, & Thimm, 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 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; there is, however, also a rigorous study of this issue (Baumann & Strass, 2022). Driven by this observation, the notion of strong equivalence has been proposed, developed and investigated in various contexts (Lifschitz, Pearce, & Valverde, 2001; Oikarinen & Woltran, 2011). Strong equivalence is a stricter version of ordinary equivalence where the semantical compliance of the given knowledge bases, even after adding novel information, is required by definition. That is, $\mathcal{F}$ and $\mathcal{G}$ are *strongly equivalent* if, for each conceivable $\mathcal{H}$, the knowledge bases $\mathcal{F} \cup \mathcal{H}$ and $\mathcal{G} \cup \mathcal{H}$ agree on their accepted conclusions. Interestingly, it is typically possible to algorithmically decide strong equivalence without computing any such set $\mathcal{H}$ explicitly, see e.g. the corresponding results for logic programs (LPs) (Lifschitz et al., 2001) or Dung-AFs (Oikarinen & Woltran, 2011).

While strong equivalence is about comparing the behavior of different knowledge bases, the *enforcement problem* (Baumann, 2012b; Wallner, Niskanen, & Järvisalo, 2017; Borg & 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, 2012b; Baumann, Rapberger, & Ulbricht, 2022; Oikarinen & Woltran, 2011). There are, however, only few studies on the aforementioned problems in structured argumentation. In structured argumentation formalisms, the arguments are constructed from a given knowledge base $\mathcal{K}$ with the goal to explicate conflicts within $\mathcal{K}$ and provide potential solutions. Thus arguments are not viewed as abstract entities, but they have some inner structure. Depending on the nature of the formalism, arguments require certain premises as well as rules in order to be derived and yield a conclusion. Prominent examples of structured argumentation formalisms are assumption-based argumentation (ABA) (Cyras, Fan, Schulz, & Toni, 2018), ASPIC$^+$ (Modgil & Prakken, 2018), defeasible logic programming (DeLP) (García & Simari, 2018), and deductive argumentation (Besnard & Hunter, 2018).

In this paper, we study the enforcement and strong equivalence problem for structured argumentation with a main focus on ABA. More specifically, our main focus is on the identification of ABA fragments where these two problems are tractable, i.e., can be decided in polynomial time. Consider for instance strong equivalence. As described above, being able to decide strong equivalence helps agents to assess whether or not two knowledge bases agree on their outcome under any conceivable expansion. Technically, awareness of strong equivalence helps to decide whether or not two knowledge bases are exchangeable in a given encoding. Intuitively, it helps understanding the represented knowledge and hints at different possibilities to express the same information. Thus, whenever strong equivalence can be decided efficiently, agents have fast access to this information. In addition, and maybe even more importantly, fragments of a formalism where strong equivalence is tractable possess properties ensuring that (in some sense) it is easy to compare knowledge bases.

While for abstract argumentation, deciding strong equivalence as well as the basic argument enforcement (Baumann, 2012b) 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: As we already mentioned, we can construct arguments from a structured argumentation formalism like ABA (Caminada, Sá, Alcântara, & Dvořák, 2015a) or logic-based argumentation (Gorogiannis & Hunter, 2011). A similar procedure also exists for logic programs (LPs) (Dung, 1995; Caminada, Sá, Alcântara, & Dvořák, 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 (Pearce, Tompits, & Woltran, 2001; Lin, 2002); here, we would hope that transferring the results from abstract argumentation will facilitate us to identify an LP fragment for which deciding strong equivalence is tractable.

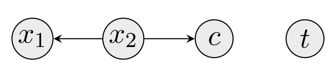
So, can we just translate a given knowledge base $\mathcal{K}$ to an argumentation graph $F_\mathcal{K}$ and solve all problems out of the box? We identify two major potential issues with this approach: (1) First, in many structured argumentation formalisms, the constructed AF $F_\mathcal{K}$ is exponential (sometimes even infinite) w.r.t. the size of the knowledge base $\mathcal{K}$. Hence, even though the problem at hand might be tractable in $F_\mathcal{K}$ this does not guarantee tractability in $\mathcal{K}$. (2) The other issues occur due to moving from static to dynamic scenarios. The obstacle is that the translation from $\mathcal{K}$ to $F_\mathcal{K}$ is tailored for static, but not for dynamical reasoning environments. We illustrate this in the following example in the context of ABA. We refer the reader to Section 2 to a formal introduction to AFs and ABA. However, the example is designed in a way that understanding all details is not necessary at this stage.

Example 1.1. *We consider an instantiation of an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$. The set $\mathcal{L} = \{c,\ \overline{c},\ m,\ t,\ \overline{t}\}$ is the set of all atoms occurring in the ABA framework and the set $\mathcal{A} = \{c, t\}$ represents so-called* assumptions*, i.e., defeasible information. Consequently, the assumptions c and t have contraries $\overline{c}$ and $\overline{t}$, respectively. We consider two rules stating that the assumption c entails m and the assumption t entails $\overline{c}$, the contrary of c:*

$$r_1 :\ m \leftarrow c. \qquad\qquad r_2 :\ \overline{c} \leftarrow t.$$

Intuitively, if we assume that c holds, then we also believe in m; if we assume t is true, then c is assumed to be false. We construct a corresponding AF F_D as follows. An AF is a directed graph consisting of nodes and directed edges representing arguments and attacked between them, respectively. Each assumption c and t induces a corresponding argument for itself; and each rule r_i yields an argument x_i. That is, x_1 is an argument for m which relies on the assumption m as premise and x_2 entails $\overline{c}$ relying on t. Attacks depend on the conclusions of arguments, e.g., x_2 attacks x_1 because $\overline{c}$ is the contrary of c. We depict F_D:

F_D :



This AF F_D represents the knowledge encoded in D, as we recall in Proposition 2.6 (Cyras et al., 2018). It turns out, however, that we have abstracted away too much information to analyze dynamic reasoning: The rule r_2 relying on t can be disabled by adding a fact stating that t is false, i.e., a rule

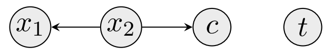
$$r_3: \ \overline{t} \leftarrow .$$

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 "$r'_2 : \overline{c} \leftarrow$", i.e., in this ABA framework $\overline{c}$ is certain and does not rely on t anymore. Hence the rules in D' are

$$r_1: \ m \leftarrow c. \qquad\qquad r'_2: \ \overline{c} \leftarrow .$$

Crucially, the instantiation procedure yields the same AF:

$F_{D'}$:



Although D and D' encode different information we obtain $F_D = F_{D'}$, i.e., the information whether $\overline{c}$ is certain or entailed from the assumption t is lost after instantiating. Hence our AFs do not carry sufficient information to investigate changes to the two ABA frameworks. Consider the following questions:

- *Is it possible to accept assumption c by adding suitable rules? The answer is affirmative in D, but negative in D'. This information cannot be extracted from F_D and $F_{D'}$.*
- *What are the stable models*[1] *after adding the fact "$\overline{t} \leftarrow .$"? In D, $\{c\}$ is then stable while in D' no assumption is acceptable after this modification. We cannot judge the situation correctly by comparing the AFs.*
- *More generally, are D and D' strongly equivalent? The answer is clearly negative when inspecting D and D' but again we cannot tell by comparing their associated AFs.*

In all of these questions, the missing piece of information is that x_2 has a hidden weakness *$\overline{t}$ 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 $\overline{t}$.*

Striving to circumvent the first issue we mentioned above regarding the size of the constructed graph $F_\mathcal{K}$, we will identify a suitable ABA fragment giving rise to only linearly many arguments when constructing the graph. To handle the second problem, we identify the minimal generalization to tailor AFs suitable for dynamic settings: (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 $(\{\alpha, \alpha \rightarrow \beta\}, \beta)$ can be the sentence $\neg\alpha$; its conclusion is β. Considering ASPIC (Modgil & Prakken, 2018), 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 $\neg d_1$.

1. A set is stable if it is conflict-free and attacks all other elements, cf. Definition 2.4.

Interestingly, the many proposals for extending Dung-AFs focus on enhancing the expressiveness of AFs, e.g. the addition of supports (Cayrol & Lagasquie-Schiex, 2005), recursive (Baroni, Cerutti, Giacomin, & Guida, 2011) as well as collective (Nielsen & Parsons, 2006) attacks, or probabilities (Thimm, 2012); hence these generalizations are not equipped with the tools necessary to investigate dynamics of structured argumentation formalisms. In contrast, our proposal does not aim at the expressive power of the AF formalism, but its capability to achieve precisely the formerly mentioned goal.

Thus, in this paper we consider a generalization of AFs by augmenting arguments with *vulnerabilities* and a *conclusion*. This formalism is indeed suitable to investigate knowledge bases that undergo changes. Notably, this approach allows us to identify a fragment of ABA for which deciding enforcement and strong equivalence becomes tractable; whereas the general case is not. As an aside, we show how our approach is flexible enough to immediately obtain similar results for LPs. Our main contributions are as follows:

- We formalize and study enforcement as well as strong equivalence for ABA. We show that, as anticipated, both problems are intractable, which is in contrast to their counterparts in abstract argumentation.
- We prove a characterization result for deciding strong equivalence for stable semantics in ABA by means of so-called SE-models, similar in spirit to research conducted in the context of LPs.
- We present our novel formalism called *conclusion and vulnerability augmented AFs (cvAFs)*. We show that cvAFs give rise to 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. This fragment consists of so-called *atomic* ABAs with *separated* contraries. We show that this fragment has the full expressive power of ABA (Proposition 5.16 and Remark 6.29)
- We transfer our results to LPs and analogously identify a fragment for which enforcement and strong equivalence is tractable.

A preliminary version of this work has been recently published (Rapberger & Ulbricht, 2022). The present study significantly extends the aforementioned paper. Most notably, we cover additional enforcement notions in Section 6 which broadens our investigation. The results regarding LPs in Section 8 are more general: We generalize the sections main Theorem 8.18 from atomic to arbitrary extension of the given programs. We also added a characterization result for strong equivalence in general ABA frameworks by means of SE-models, similar in spirit to SE-models for LPs. Moreover, besides presenting all required proofs in full details, we provide a more comprehensive selection of examples throughout the paper. Needless to say, the absence of space limits gives us the chance to better put our results in context, give a stronger intuition about our formal technicalities, and discuss related work in more detail.

2. Background

Abstract Argumentation. We fix a countably infinite background set U. An argumentation framework (AF) (Dung, 1995) is a directed graph $F = (A, R)$ where $A \subseteq U$ represents a set of arguments and $R \subseteq A \times A$ models *attacks* between them. For two AFs $F = (A, R)$ and $G = (B, S)$, we define their union $F \cup G = (A \cup B, R \cup S)$. 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 σ 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 $E^+ = A \setminus E$.*

We will sometimes make use of the characteristic function Γ_F of an AF F, defined as $\Gamma_F(E) = \{a \in A \mid E \text{ defends } a\}$. If clear from context, we omit the subscript F.

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 $a_0 \leftarrow a_1, \ldots, a_n$, where $a_i \in \mathcal{L}$ for all $i \leq n$, $head(r) = a_0$ 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 $^- : \mathcal{A} \to \mathcal{L}$.*

Assumption 2.3. *In this work, we focus on ABA frameworks $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ that are*

- flat*, i.e., $head(r) \notin \mathcal{A}$ for each rule $r \in \mathcal{R}$; and*
- finite*, i.e., i.e., $\mathcal{L}$, $\mathcal{R}$, $\mathcal{A}$ are finite; moreover,*
- *$\mathcal{L}$ is a set of atomic formulas; and*
- *each rule $r \in \mathcal{R}$ is stated explicitly (given as input).*

The restriction to flat frameworks is widely used. It allows for a clear distinction between supporting elements (assumptions) and derivable statements. We assume further restrictions (items 2 to 4) due to our complexity-theoretic analysis. Note that by item 4, we consider only ABA frameworks with ground rules, that is, no rule contains any variables.

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 of T 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 of T 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 $body(r) = \emptyset$.

We note that each assumption derives itself via $\{a\} \vdash_{\emptyset} a$ (a tree with no internal nodes). For a set S of assumptions, we let $\overline{S} = \{\overline{a} \mid a \in S\}$. We denote by

$$Th_D(S) = \{p \mid \exists S' \subseteq S : S' \vdash_R p\}$$

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.

A set of assumptions S *attacks* a set of assumptions T if there is some $a \in T$ s.t. $\overline{a} \in Th_D(S)$. If S attacks $\{a\}$, we simply say S attacks a. S is conflict-free, $S \in cf(D)$, if it does not attack itself. A conflict-free set S is admissible, $S \in ad(D)$, if it defends itself. We recall complete, grounded, preferred, and stable semantics (abbr. *co*, *gr*, *pr*, *stb*).

Definition 2.4. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework and let $S \in ad(D)$. Then*

- *$S \in co(D)$ 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 $x \in \mathcal{A} \setminus S$.*

For a semantics $\sigma \in \{co, gr, pr, stb\}$, we define $\sigma_{Th}(D) = \{Th_D(S) \mid S \in \sigma(D)\}$.

We say that an assumption $a \in \mathcal{A}$ (atom $p \in \mathcal{L}$) is *credulously accepted* w.r.t. 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 2.5. *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 $p \in \overline{S'}$.*

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 (Cyras et al., 2018)).

Proposition 2.6. *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 $\{S' \vdash p \mid \exists S' \subseteq S, R \subseteq \mathcal{R} : S' \vdash_R p\} \in \sigma(F)$.*

Computational Complexity. We assume the reader to be familiar with the basic concepts of computational complexity theory (Arora & Barak, 2009; Papadimitriou, 1994)- As usual, by P (polynomial time) we denote the class of all problems which can be solved via a deterministic polynomial-time Turing machine. As usual, we will call these problems *tractable*. By NP we denote the class of all problems which can be solved via a (not necessarily deterministic) polynomial-time turing machine and coNP is the complementary class to NP. We call problems which are hard for NP or coNP *intractable* (since, according to standard complexity assumptions, they cannot be solved in polynomial time).

The canonical NP-complete problem is Sat (satisfiability): Given a propositional formula Φ over a set X of atoms in 3-CNF, i.e., Φ can be interpreted as a set of clauses, the output is "yes" if and only if there is a satisfying assignment $\omega : X \to \{0, 1\}$ for Φ. Analogously, Φ as above is a "yes" instance of Unsat (unsatisfiability) if and only if there is no satisfying assignment ω. The problem Unsat is the prototypical coNP-complete problem.

3. Warm-Up: Enforcement and Strong Equivalence for Dung-AFs

This paper is driven by the observation that research on dynamics in abstract argumentation theory cannot be applied to structured argumentation directly. Nonetheless, this research lays relevant foundations for and significantly inspires our investigation. We will develop similar techniques, tailored to the needs of our setting. It is therefore important to be aware of the most relevant research for abstract AFs.

3.1 Enforcement

Suppose we are involved in some discussion where our point of view stands to lose. Naturally, we seek for possibilities to bring forward further arguments which support our claims. Formalizing situations of this kind leads to the so-called *enforcement* problem. In the context of abstract argumentation, the most basic version (Baumann, 2012b) addresses the issue of ensuring acceptance of a certain target set of arguments. That is, given some AF $F = (A, R)$ and an arguments $S \subseteq A$, is it possible to move to some super-framework $F \cup H$ (i.e., add arguments and attacks) s.t. S is credulously accepted in the updated $F \cup H$? Formally, we obtain the following basic enforcement notion.

Definition 3.1. *Let $F = (A, R)$ be an AF, σ any semantics, and $S \subseteq A$. We say S is* enforceable *if there is some AF G s.t. there is some σ-extension $E \in \sigma(F \cup G)$ s.t. $S \subseteq E$.*

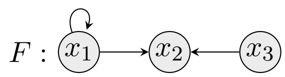
It has been formally established that this is possible for any set S of arguments which does not have any internal conflict. Thereby, we can even restrict our attention to so-called normal expansions, as formalized next.

Definition 3.2. *Let $F = (A, R)$ be an AF. We call $G = (A \cup A', R \cup R')$ an* expansion *of F; G is a* normal expansion *if $(a, b) \in R'$ implies $a \in A'$ or $b \in A'$.*

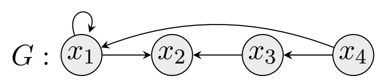
Intuitively, G is a normal expansion if each novel attack involves at least one newly added argument. Applying (Baumann & Brewka, 2010, Theorem 4) to our semantics we obtain the following result implying that any conflict-free set S can be enforced by means of a normal expansion.

Theorem 3.3. *Let $F = (A, R)$ be an AF and $S \subseteq A$. Let $\sigma \in \{ad, co, gr, pr, stb\}$. There is a normal expansion G of F s.t. $S \subseteq E$ for some extension $E \in \sigma(G)$ iff $S \in cf(F)$.*

Example 3.4. *Let F be given as follows.*



Let $S = \{x_2\}$. Since x_2 is not self-attacking we can enforce it by counter-attacking both x_1 and x_3. We introduce a suitable x_4. This corresponds to a normal expansion G of F:



We have $\{x_2, x_4\} \in \sigma(G)$ for all semantics considered in this paper; thus acceptance of x_2 is indeed ensured.

3.2 Strong Equivalence

The idea behind strong equivalence is to develop a notion of equivalence which is robust even in dynamic scenarios, i.e., when the given knowledge bases undergo changes. Therefore, two AFs F and G are defined to be strongly equivalent (Oikarinen & Woltran, 2011) if they output the same extensions even under additional information.

Definition 3.5. *Two AFs F and G are* strongly evquivalent *w.r.t. a semantics σ (denoted by $F \equiv_s^\sigma G$) if and only if $\sigma(F \cup H) = \sigma(G \cup H)$ holds for each conceivable AF H.*

That is, the two given AFs are evaluated equivalently under σ even if we are faced with new arguments and attacks formalized in H. Put differently, if we view $F \cup H$ as a single AF, then we can replace F with the strongly equivalent G and obtain $G \cup H$ possessing the same σ-extensions.

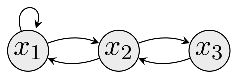
When merely inspecting the definition of strong equivalence, it appears to be a computationally hard task at first glance: After all, any conceivable H has to be checked. It turned out, however, that strong equivalence for AFs can be characterized by verifying the syntactical identity of so-called (semantics-dependent) kernels. We want to stress that these kernels are obtained by suitable modifications to the attack relation of the given AFs and do not remove or add any arguments.

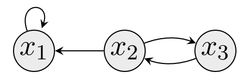
Let us recall the kernels for stable, admissible, complete, and grounded semantics (Oikarinen & Woltran, 2011).

Definition 3.6. *For an AF $F = (A, R)$, we define the* stable kernel $F^{sk} = (A, R^{sk})$*;* admissible kernel $F^{ak} = (A, R^{ak})$*; the* complete kernel $F^{gk} = (A, R^{gk})$ *and the* grounded kernel $F^{gk} = (A, R^{gk})$ *with*

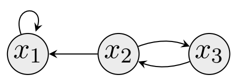
$$
\begin{aligned}
R^{sk} &= R \setminus \{(a, b) \mid a \neq b, (a, a) \in R\} \\
R^{ak} &= R \setminus \{(a, b) \mid a \neq b, (a, a) \in R, \{(b, a), (b, b)\} \cap R \neq \emptyset\}; \\
R^{ck} &= R \setminus \{(a, b) \mid a \neq b, (a, a), (b, b) \in R\}; \\
R^{gk} &= R \setminus \{(a, b) \mid a \neq b, (b, b) \in R, \{(b, a), (a, a)\} \cap R \neq \emptyset\}.
\end{aligned}
$$

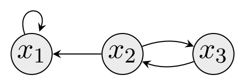
Example 3.7. *Consider the following AFs F and G:*

F : 

G : 

When computing the stable kernel F^{sk} we remove outgoing attacks of self-attacking arguments. Hence, for F the attack (x_1, x_2) does not occur in F^{sk}. The stable kernel of G coincides with G since there is nothing to remove. We obtain

F^{sk} : 

G^{sk} : 

Observe the intuition behind the kernel: Since x_1 can never occur in a stable extension, it must be defeated *by any $E \in stb(F)$. Hence the outgoing attack of x_1 is irrelevant.*

These kernels serve to characterize strong equivalence as we recall next.

Theorem 3.8 ((Oikarinen & Woltran, 2011))**.** *For any two AFs F and G,*

$$\begin{aligned} F \equiv_s^{stb} G &\text{ iff } F^{sk} = G^{sk} \\ F \equiv_s^{\sigma} G &\text{ iff } F^{ak} = G^{ak} \text{ for } \sigma \in \{ad, pr\} \\ F \equiv_s^{co} G &\text{ iff } F^{ck} = G^{ck} \\ F \equiv_s^{gr} G &\text{ iff } F^{gk} = G^{gk} \end{aligned}$$

We write $F^{k(\sigma)}$ to denote the kernel which characterize strong equivalence w.r.t. σ.

Example 3.9. *Heading back to F and G we saw in the previous example that the two stable kernels coincide; i.e., $F^{sk} = G^{sk}$. We deduce $F \equiv_s^{stb} G$.*

Note that computing and comparing two kernels is a simple computational task. Hence deciding strong equivalence for AFs is tractable for all semantics considered in this paper.

3.3 Limitations

When inspecting the proof techniques for the enforcement (Baumann, 2012b) and strong equivalence (Oikarinen & Woltran, 2011) results it becomes apparent that they heavily rely on the abstract nature of the arguments. More specifically, it is usually assumed (and oftentimes used) that a novel argument can simply attack anything within the already given AFs. This is not only somewhat questionable from an intuitive point of view, but also makes it hard to apply these results to AFs which stem from instantiating some knowledge base.

To illustrate this, suppose Jane and Antoine discuss their plans for the weekend. Formally speaking, they exchange arguments yielding a certain AF at any time. Bringing forward further arguments naturally induces expansions of the currently given one. Naturally, both argue in favor of the outcome they prefer, i.e., their ultimate goal is to enforce a certain argument. Jane would like to go to the cinema since she got recommended this new blockbuster about dynamic reasoning. She therefore brings forward the argument

x_1 : "If we go to the cinema, we can watch the movie I heard about."

Antoine does not want to go to the cinema, because he is not sure whether he is interested in this movie. Looking for an excuse, he brings forward the following argument:

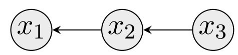
x_2 : "If the tickets are too expensive, I would prefer not to go to the cinema."

However, Jane got a cinema voucher for her last birthday. She happily points out:

x_3 : "No worries, we can go there for free."

If we depict this simple exchange of arguments as a Dung-AF we obtain the following graph $F = (A, R)$.

F :



Since Antoine was not honest about his reasoning, he is not satisfied with the way this discussion went and aims to enforce his argument x_2. According to (Baumann, 2012b, Theorem 4), he could achieve this by bringing forward some novel argument which defeats x_3. However, it is clear that x_3 is a fact in this context, i.e., there is no reasonable argument against it.

Needless to say, in a different context it might be possible for Antoine to enforce his argument x_2. Hence the question arises under which conditions it is indeed possible for him to achieve his goal? The attentive reader may already anticipate that this question cannot be answered in the context of abstract Dung-AFs, since here it would always be possible to add new incoming attacks to any argument.

Our approach is therefore to start our investigation from the point of view of the structured setting: Given a knowledge base, adding new information induces certain changes in the instantiated AF. Guided by the possible modifications, we can work out a suitable abstract framework which abides by the given restrictions. Afterwards, we demonstrate how to translate our results back to the knowledge base.

4. Dynamics in Assumption-based Argumentation

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.

The *expansion* of a framework is a central concept to both of our problems: naturally, expansions are an integral part of strong equivalence; moreover, since we assume that existing knowledge cannot be deleted, we study claim enforcement under the assumption that we can only add novel elements to our knowledge representation formalism. Below, we settle the notion of framework expansions for ABA frameworks. We fix $\mathcal{L}$ and a countably infinite set of assumptions $\mathcal{L}_A \subseteq \mathcal{L}$ and the contrary function $^- : \mathcal{L}_A \to \mathcal{L}$. We consider expansions component-wise.

Definition 4.1. *Let* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ *and* $D' = (\mathcal{L}, \mathcal{R}', \mathcal{A}', ^-)$ *be ABA frameworks with* $\mathcal{A}, \mathcal{A}' \subseteq \mathcal{L}_A$ *and* $^- : \mathcal{L}_A \to \mathcal{L}$*. We call*

$$D \cup D' := (\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A} \cup \mathcal{A}', ^-)$$

the expansion *of* D *by* D'*.*

For a rule $r = p \leftarrow S$, we write $D \cup \{r\}$ short for $D \cup D'$ with $D' = (\mathcal{L}, \{r\}, \emptyset, ^-)$. For a set of assumptions $\mathcal{A}' \subseteq \mathcal{L}_A$, we write $D \cup \mathcal{A}'$ for $D \cup D'$ with $D' = (\mathcal{L}, \emptyset, \mathcal{A}', ^-)$.

We note that by fixing the set of sentences $\mathcal{L}$ and the contrary function over a fixed set of assumptions, we avoid the case that framework expansions are not compatible. Moreover, in this way we guarantee that the expansion $D \cup D'$ is flat as well (recall that by Assumption 2.3, we focus exclusively on flat ABA frameworks throughout this work). In what follows, we assume that for all considered ABA frameworks $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, it holds that $\mathcal{A} \subseteq \mathcal{L}_A$, and $^- : \mathcal{L}_A \to \mathcal{L}$.

4.1 Conclusion Enforcement

We require that a conclusion p cannot be enforced by simply adding conclusion p or elements that introduce a novel argument with conclusion p since this would trivialize the problem. Formally, we consider the following problem:

Definition 4.2. *Given an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, a conclusion $p \in \mathcal{L}$, and a semantics σ, we say that p is* enforceable with respect to σ *iff there is some expansion $D \cup D'$ (and p does not appear as a head in D') such that there is $S \in \sigma_{Th}(D \cup D')$ with $p \in S$ (we say, p is credulously accepted with respect to σ in $D \cup D'$).*

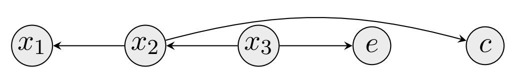
Since assumptions are contained in the conclusion-extensions of an ABA framework, we cover with this notion both conclusion- and assumption-enforceability.

We observe an interesting discrepancy between structured and abstract formalisms: While it is possible to credulously enforce any argument in a given AF as long as it is not self-attacking, the problem of claim enforceability is more involved.

Example 4.3. *Let us formalize the discussion between Jane and Antoine and why Antoine loses the argument. We construct the ABA framework $D = \{\mathcal{L}, \mathcal{A}, \mathcal{R}, ^-\}$ with the occurring atoms cinema (c), $\overline{\text{cinema}}$ ($\bar{c}$), expensive (e), $\overline{\text{expensive}}$ ($\bar{e}$), and movie (m):*

$$\mathcal{L} = \{c,\ \bar{c},\ e,\ \bar{e},\ m\} \qquad \mathcal{A} = \{c,\ e\} \qquad \mathcal{R} = \{r_1 : m \leftarrow c., \quad r_2 : \bar{c} \leftarrow e., \quad r_3 : \bar{e} \leftarrow .\}.$$

If we instantiate the corresponding AF F_D we get the following graph (with rule r_i inducing argument x_i and the assumptions c and e corresponding ones).

F_D : 

In order for Antoine to win his argument, we would have to enforce x_2. However, x_3 (stemming from r_3) corresponds to a fact and therefore no expansion $D \cup D'$ of D would achieve this. Meanwhile, when inspecting the graph F_D, x_2 can be simply enforced by adding an attack against x_3 (Baumann, 2012b, Theorem 4). So we see that the abstract point of view does not yield the desired outcome.

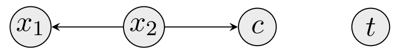
Now suppose Antoine uses the following argument x_2' instead:

> *x_2, revised: "If I do not like the trailer of this movie, I would prefer not to go to the cinema."*

Jane's voucher is not a counter-argument anymore and we end up with the following ABA $D' = \{\mathcal{L}', \mathcal{A}', \mathcal{R}', \overline{}\}$ *where* t *is the abbreviation for Antoine not liking the trailer:*

$$\mathcal{L}' = \{c,\ \overline{c},\ m,\ t,\ \overline{t}\} \qquad \mathcal{A}' = \{c,\ t\} \qquad \mathcal{R}' = \{r_1 : m \leftarrow c., \quad r_2' : \overline{c} \leftarrow t.\}.$$

This time, we obtain the following AF (where we do not change the naming convention, i.e., the second rule induces x_2*)*

$F_{D'}$:


Now Antoine's argument is accepted, but Jane's x_1 *can be enforced when Antoine likes the trailer (i.e.,* $\overline{t}$ *holds). Indeed, if we consider the ABA* $H = (\{\overline{t}, \emptyset, \{\overline{t} \leftarrow .\}, \emptyset\})$*, then with* $r = \overline{t} \leftarrow .$ *the corresponding expansion* $D' \cup H = \{\mathcal{L}', \mathcal{A}', \mathcal{R}' \cup \{r\}, \overline{}\}$ *is given as*

$$\mathcal{L}' = \{c,\ \overline{c},\ m,\ t,\ \overline{t}\} \quad \mathcal{A}' = \{c,\ t\} \quad \mathcal{R}' \cup \{r\} = \{r_1 : m \leftarrow c., \quad r_2' : \overline{c} \leftarrow t., \quad r : \overline{t} \leftarrow .\}.$$

Here one can verify directly that m *is accepted; hence Jane's wish is enforced.*

This example already hints at the fact that enforcement in ABA involves identifying suitable claims one could add to the knowledge base. As it turns out, enforcement in ABA is indeed NP-hard.

Reduction 4.4. *For a CNF formula* φ *with clauses* $C = \{c_1, \ldots, c_n\}$ *over variables in* X*, we define the corresponding ABA framework* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ *with*

- $\mathcal{A} = \{x_a^T, x_p^F, x_a^F, x_p^T \mid x \in X\} \cup \{c, e\}$ *where*
- $\overline{x_p^F} = x_a^T$, $\overline{x_p^T} = x_a^F$, *and* $\overline{c}, \overline{e}, \overline{x_a^T}, \overline{x_a^F} \in \mathcal{L} \setminus \mathcal{A}$.

Moreover, $\mathcal{R}$ *contains the following rules:*

- $\varphi \leftarrow c, e$,
- *for all* $x \in X$, $\mathcal{R}$ *contains a rule* $\overline{e} \leftarrow x_p^T, x_p^F$;
- *for each* $i \leq n$, $\mathcal{R}$ *contains a rule of the form* $\overline{c} \leftarrow \{x_a^T \mid x \in c_i\} \cup \{x_a^F \mid \neg x \in c_i\}$.

For each variable, we introduce four assumptions, associated to different truth values on the one hand, and to 'active' (x_a^T, x_a^F) and 'passive' (x_p^T, x_p^F) assumptions on the other hand, meaning that the 'passive' assumptions cannot be defeated by newly introduced rules because their contrary is itself an assumption (recall that we are operating in flat frameworks). Figure 1 depicts the resulting AF for the formula $(x \vee y) \wedge (\neg x) \wedge (\neg y)$.

Theorem 4.5. *Deciding whether a conclusion* p *(assumption* a*) is enforceable in a given ABA framework* D *w.r.t. a semantics* $\sigma \in \{gr, co, pr, stb\}$ *is* NP*-hard.*

Sketch of Proof. We present a reduction from SAT which shows hardness for grounded, complete, preferred, and stable semantics. Given a CNF formula φ with clauses $C = \{c_1, \ldots, c_n\}$ over variables in X, we let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be defined as in Reduction 4.4.

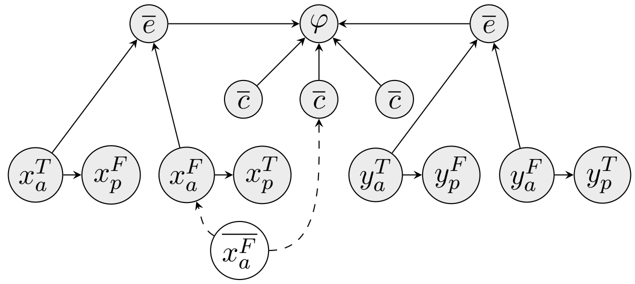


Figure 1: Reduction from the Proof of Theorem 4.5 for the formula φ given by clauses $\{x, y\}, \{\neg x\}, \{\neg y\}$; depicted with the argument arising from the additional rule $\overline{x_a^F} \leftarrow$ (fact), in white, with dashed attacks.

It holds that φ is enforceable w.r.t. σ iff φ is satisfiable: The reason is that almost all relevant elements in the construction are assumptions and since we need to stay flat, we can only add rules with $\overline{x_a^T}$ resp. $\overline{x_a^F}$ in the head in order to manipulate whether or not φ is accepted. This simulates the search for a satisfying assignment. Moving to the setting of assumption-enforcement requires some technical adaptation to the construction, but the overall idea is analogous. More details can be found in Appendix A. □

To summarize, enforcement as defined above is intractable for ABA, although quite straightforward in the AF case. Note that this happens even though we considered a natural translation of the very basic notion of enforcement into the realm of ABA. The intuitive reason is that the abstract investigation of enforcement is free to consider any conceivable new argument, while we have to abide by rules imposed by the given knowledge base.

4.2 Strong Equivalence

In this section, we discuss strong equivalence for ABA. Notice that we consider strong equivalence relative to different fragments of ABA.

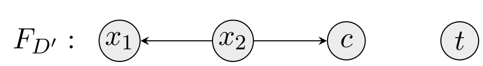
Definition 4.6. *Consider a fragment $\mathfrak{C}$ of ABA frameworks. Two ABA frameworks $D, D' \in \mathfrak{C}$ are* strongly equivalent *to each other with respect to a semantics σ iff*

1. *$\sigma(D \cup H) = \sigma(D' \cup H)$ for each $H \in \mathfrak{C}$; and*
2. *$D \cup \mathcal{H}$ and $D' \cup \mathcal{H}$ are instances of $\mathfrak{C}$.*

Example 4.7. *Consider the situation after Antoine changed his argument to x_2', i.e., we have $D' = \{\mathcal{L}', \mathcal{A}', \mathcal{R}', ^{-}\}$ with*

$$\mathcal{L}' = \{c,\ \overline{c},\ m,\ t,\ \overline{t}\} \qquad \mathcal{A}' = \{c,\ t\} \qquad \mathcal{R}' = \{r_1 : m \leftarrow c., \quad r_2' : \overline{c} \leftarrow t.\}$$

and corresponding AF



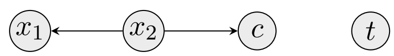
As a final variation of this example, suppose Antoine just says he doesn't want to go to the cinema without any condition, i.e., he simply claims

> x_2*, again revised: "I will watch the trailer, but I prefer not to go to the cinema."*

This yields D'' *with*

$$\mathcal{L}'' = \{c,\ \bar{c},\ m,\ t,\ \bar{t}\} \qquad \mathcal{A}'' = \{c,\ t\} \qquad \mathcal{R}'' = \{r_1 : m \leftarrow c., \quad r_2'' : \bar{c} \leftarrow .\}$$

Since at this point there is no counter-argument to t *anyway, this does not make any difference for the underlying AF:*

$F_{D''}$:


Since $F_{D'}$ *and* $F_{D''}$ *coincide, they are trivially strongly equivalent. However, this is not true for the underlying ABA frameworks which is shown by our previous expansion with* $H = (\{\bar{t}, \emptyset, \{\bar{t} \leftarrow .\}, \emptyset\})$*: In* $D' \cup H$*, the atom* $\bar{c}$ *is rejected; in* $D'' \cup H$ *it is accepted (*$\bar{c}$ *is a fact in* D''*).*

As the attentive reader might already anticipate, we infer intractability again. By adapting the proof of Theorem 4.5 we obtain the following result (cf. Appendix A).

Theorem 4.8. *Deciding whether two ABA frameworks are strongly equivalent w.r.t. a given semantics* $\sigma \in \{gr, co, pr, stb\}$ *is* coNP*-hard.*

As in the case of enforcement, this is in contrast to the (syntactical and tractable) kernel characterizations of strong equivalence for the AF case (Oikarinen & Woltran, 2011). Again, the reason for this discrepancy can be identified by making an analogous observation: Strong equivalence in the AF case can be decided by a simple syntactical characterization because the many possible ways to extend the two given AFs will almost always yield some way to distinguish them. This is why the answer is almost always negative and the procedure from (Oikarinen & Woltran, 2011) identifies the rare cases where non-identical AFs are strongly equivalent.

In our setting, however, the given knowledge base restricts the possible expansions and their impact on the updated knowledge base. This is why the possible modifications are harder to identify and the problem becomes intractable. Our analysis in this section also reveals that the research conducted for dynamics in Dung-AFs cannot be applied directly to the instantiated AF. This observation is the driving force for the subsequent sections.

4.3 Excursus: ABA SE-Models

Strong equivalence for logic programs under stable model semantics has been characterized using the Logic of Here and There (Lifschitz et al., 2001) and later simplified by means of so-called SE-models (Turner, 2001). Since ABA is known to be closely related to logic programming (Sá & Alcântara, 2021), the question arises whether there is a similar characterization for ABA frameworks under stable semantics. In this subsection, before moving on to developing our instantiation technique for dynamics, we will answer this question affirmatively.

Both our notion of SE models as well as the proof of the strong equivalence characterization are similar in spirit to their LP counterparts. In LPs, semantics are constructed by guessing a suitable set of atoms and then constructing the *reduct* of the given program w.r.t. the candidate set. Let us first settle how we can proceed analogously for ABA frameworks.

To this end we use the notion of so-called *candidate sets*. Intuitively, a candidate set X corresponds to a set of atoms in $\mathcal{L}$ which we view as accepted. Assumptions only occur in a candidate set whenever they are the *contrary* of some other assumption. Thus, candidate sets consist of all kinds of atoms $p \in \mathcal{L} \setminus A$ and assumptions $a \in \mathcal{A}$ whenever necessary in order to encode some contrary.

Definition 4.9. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework. A set $X \subseteq \mathcal{L}$ is called a* candidate set *if $a \in X$ for some $a \in \mathcal{A}$ implies $a \in \overline{\mathcal{A}}$. A candidate set X is* conflict-free *if there is no assumption $a \in X$ with $\overline{a} \in X$.*

Example 4.10. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be the ABA framework where $\mathcal{L} = \{a, b, p, q, \overline{b}\}$, $\mathcal{A} = \{a, b\}$, and $\overline{a} = b$. The set $X = \{p, q, \overline{b}\}$ is a conflict-free candidate set which can be seen since X does not contain any assumption; $Y = \{b\} = \{\overline{a}\}$ is a candidate set since b is a contrary (the contrary of a); $Z = \{p, q, a\}$ is no candidate set.*

Now the decisive step is to construct the reduct D^X of an ABA framework which partially evaluates X and then returns some ABA framework without any assumptions left.

Definition 4.11 (ABA reduct)**.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework and $X \subseteq \mathcal{L}$ a candidate set. The* reduct *of D w.r.t. X is the ABA framework $D^X = (\mathcal{L}, \mathcal{R}^X, \emptyset, \emptyset)$ where the set $\mathcal{R}^X$ of rules is given as*

$$\mathcal{R}^X = \{head(r) \leftarrow body(r) \setminus \mathcal{A} \mid r \in \mathcal{R},\ \overline{\mathcal{A} \cap body(r)} \cap X = \emptyset\}$$
$$\cup \{a \leftarrow . \mid a \in X \cap \mathcal{A}\}.$$

Observe in particular that D^X for some candidate set X does not contain any assumptions (although it contains atoms which are assumptions in the initial ABA framework D). Note how D^X corresponds to evaluating the assumptions in D according to X: rules relying on negated assumptions get removed, whereas the assumptions in X are added as facts, i.e., they are not considered defeasible anymore in D^X. Now we define the notion of a model of a framework without any assumption.

Definition 4.12 (Model)**.** *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework with $\mathcal{A} = \emptyset$. A candidate set X is a model of D, denoted by $X \vDash D$, if $Th_D(\emptyset) \subseteq X$.*

Now we can evaluate D in a two-step procedure:

- Guess some conflict-free candidate set X;
- compute D^X and check whether $X \vDash D^X$ holds.

Example 4.13. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be the ABA framework where $\mathcal{L} = \{a, b, p, q, \overline{b}\}$, $\mathcal{A} = \{a, b\}$, and $\overline{a} = b$. Moreover let $\mathcal{R}$ be the set*

$$\mathcal{R}: \qquad p \leftarrow a. \qquad q \leftarrow p. \qquad \overline{b} \leftarrow q.$$

of rules. Consider the candidate set $X = \{p, q, \bar{b}\}$*. The reduct w.r.t.* X *contains the rules*

$$\mathcal{R}^X: \qquad p \leftarrow . \qquad q \leftarrow p. \qquad \bar{b} \leftarrow q.$$

In particular, $X \models D^X$ *since* $Th_{D^X}(\emptyset) = \{p, q, \bar{b}\}$*. Moreover,* $\{a\} \in stb(D)$ *with* $Th_D(\{a\}) = X \cup \{a\}$ *corresponding to the candidate set. Analogously, let* $Y = \{\bar{a}\}$ *(which is the assumption b). Then* D^Y *contains the rules*

$$\mathcal{R}^Y: \qquad q \leftarrow p. \qquad \bar{b} \leftarrow q. \qquad \bar{a} \leftarrow .$$

with $Th_{D^Y}(\emptyset) = \{\bar{a}\} = Y$*, i.e.,* $Y \models D^Y$*. Note that* $\{b\} \in stb(D)$ *as well.*

Indeed, candidate sets and their relation to the corresponding reduct are related to stable extensions in D. Before proving this, we require the following auxiliary lemma.

Lemma 4.14. *Let* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ *be an ABA framework, let* X *be a conflict-free candidate set, and let* $S = \{a \in \mathcal{A} \mid \bar{a} \notin X\}$*. Then* $Th_D(S) \subseteq Th_{D^X}(\emptyset)$*.*

Proof. Let $p \in Th_D(S)$. Then, $S \vdash_R p$ for some set R of rules where $r \in R$ implies $body(r) \cap \{a \in \mathcal{A} \mid \bar{a} \in X\} = \emptyset$. That is, $\overline{body(r) \cap \mathcal{A}} \cap X = \emptyset$. Hence for each $r \in R$ we have $head(r) \leftarrow body(r) \setminus \mathcal{A}$ occurring in $\mathcal{R}^X$. Therefore, the same inference can be done in D^X, even without any assumption. Thus $p \in Th_{D^X}(\emptyset)$. □

The relation between candidate sets and stable extensions is as follows.

Proposition 4.15. *Let* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ *be an ABA framework.*

- *Let* X *be a conflict-free candidate set. If* X *is a minimal model of* D^X*, then for* $S = \{a \in \mathcal{A} \mid \bar{a} \notin X\}$ *we have* $S \in stb(D)$*.*
- *If* $S \in stb(D)$*, then* $X = Th_D(S) \setminus \{a \in \mathcal{A} \mid a \notin \overline{\mathcal{A}}\}$ *is a conflict-free candidate set forming the minimal model of* D^X*.*

Proof.

- Suppose X is a as described and let $S = \{a \in \mathcal{A} \mid \bar{a} \notin X\}$.

  (conflict-free) Assume $\bar{a} \in Th_D(S)$ for some $a \in \mathcal{A}$. By Lemma 4.14 $\bar{a} \in Th_{D^X}(\emptyset)$. Since X is a model of D^X we get $Th_{D^X}(\emptyset) \subseteq X$. Consequently $\bar{a} \in X$ and thus, $a \notin S$ by choice of S.

  (stable) Now suppose $a \notin S$. By choice of S we must have $\bar{a} \in X$.

  - First suppose $\bar{a}$ is no assumption, i.e., $\bar{a} \in \mathcal{L} \setminus \mathcal{A}$. Since X is a minimal model of D^X we deduce $\bar{a} \in Th_{D^X}(\emptyset)$. Hence $\bar{a}$ is derivable from the rules satisfying $\overline{body(r) \cap \mathcal{A}} \cap X = \emptyset$, i.e., $\bar{a} \in Th_D(S)$ again by choice of S.
  - Now let $\bar{a} \in \mathcal{A}$. Since X is a conflict-free candidate set, the contrary of $\bar{a}$ does not occur in X. Hence by construction of S we have $\bar{a} \in S$.

- Let $S \in stb(D)$.

  (candidate set) The set $X = Th_D(S) \setminus \{a \in \mathcal{A} \mid a \notin \overline{\mathcal{A}}\}$ contains only assumptions which in turn are contrary of other assumptions.

  (conflict-free) Since S is stable, $Th_D(S)$ does not contain $\overline{a}$ whenever $a \in S$.

  (model of D^X) Let $p \in Th_{D^X}(\emptyset)$. If $p \in \mathcal{A}$, then $p \in X$ by construction of D^X. Otherwise, p can be entailed from the set $\{r \in \mathcal{R} \mid \overline{body(r) \cap \mathcal{A}} \cap X = \emptyset\}$ of rules, i.e., from rules whose body elements contain no assumption in $\overline{S}$. By definition of stable semantics, $p \in Th_D(S)$ for all atoms $p \in \mathcal{L}$ that only rely on the above set of rules. Since we assume p to be no assumption, $p \in X$ follows.

  (minimal model) If X and S are as above, then in turn $S = \{a \in \mathcal{A} \mid \overline{a} \notin X\}$ and thus, Lemma 4.14 is applicable. We get

  $$X \setminus \mathcal{A} = Th_D(\emptyset) \setminus \mathcal{A} \subseteq Th_{D^X}(\emptyset) \setminus \mathcal{A}$$

  and thus the claim for each $p \in X \setminus \mathcal{A}$. The case $p \in \mathcal{A}$ is by construction of D^X. □

Now let us turn our attention towards dynamic scenarios. First, the notion of candidate sets and models of ABA frameworks is compatible with the union of ABAs in the following sense.

Proposition 4.16. *Let D and D' be two ABA frameworks. If X is a candidate set, then*

- $(D \cup D')^X = D^X \cup (D')^X$,
- $X \models (D \cup D')^X$ *iff* $X \models D^X$ *and* $X \models (D')^X$.

Proof. This is clear. □

Moreover, the bigger the candidate set X, the fewer atoms can be entailed from D^X. However, observe that this is not the case for assumptions occurring in the candidate set since they are added to the respective reduct as facts. Formally, we have the following relation.

Proposition 4.17. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework and $X \subseteq Y$ two candidate sets. Then $Th_{D^Y}(\emptyset) \setminus \mathcal{A} \subseteq Th_{D^X}(\emptyset) \setminus \mathcal{A}$.*

Proof. Let $p \in Th_{D^Y}(\emptyset)$ s.t. p is no assumption. Hence p can be entailed from the set

$$\mathcal{R}^Y = \{head(r) \leftarrow body(r) \setminus \mathcal{A} \mid r \in \mathcal{R},\ \overline{\mathcal{A} \cap body(r)} \cap Y = \emptyset\}$$

of rules. Clearly,

$$\begin{aligned}\mathcal{R}^Y &= \{head(r) \leftarrow body(r) \setminus \mathcal{A} \mid r \in \mathcal{R},\ \overline{\mathcal{A} \cap body(r)} \cap Y = \emptyset\} \\ &\subseteq \{head(r) \leftarrow body(r) \setminus \mathcal{A} \mid r \in \mathcal{R},\ \overline{\mathcal{A} \cap body(r)} \cap X = \emptyset\} \\ &= \mathcal{R}^X,\end{aligned}$$

i.e., p can be entailed from $\mathcal{R}^X$ as well. We deduce $p \in Th_{D^X}(\emptyset) \setminus \mathcal{A}$. □

Observe that in the previous proposition, $\mathcal{A}$ is the set of assumptions occurring in the initial ABA framework. Hence when removing $\mathcal{A}$ from $Th_{D^Y}(\emptyset)$, we remove those atoms which are assumptions in D; D^Y itself does not contain any assumption by construction of the reduct.

Now let us define SE-models of ABA frameworks. With the notations we have established, they are similar to SE-models for LPs.

Definition 4.18. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework. Let X and Y be two conflict-free candidate sets. The tuple (X, Y) is called an* SE-model *for D if*

- $X \subseteq Y$,
- $X \models D^Y$,
- $Y \models D^Y$.

If (X, Y) is an SE-model of D and $X \neq Y$, then $X \subsetneq Y$ and hence, Y is no minimal model of D^Y. On the other hand, if (Y, Y) is the only SE-model with Y in the second component, then Y is indeed a minimal model of D^Y. Hence given Proposition 4.15, a candidate set Y corresponds to some stable extension of D iff (Y, Y) is an SE-model and there is no other SE-model of D of the form (X, Y) with $X \neq Y$.

Next we show that SE-models indeed characterize strong equivalence of ABA frameworks. We require one further auxiliary results before proving the main theorem of this subsection.

Lemma 4.19. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework and (X, Y) be an SE-model of D. Then $X \cap \mathcal{A} = Y \cap \mathcal{A}$.*

Proof. Due to $X \subseteq Y$, the $\subseteq$-direction is immediate. For $(\supseteq)$, suppose $a \in Y$ is an assumption. Then $a \in Th_{D^Y}(\emptyset)$ and hence, $X \models D^Y$ implies $a \in X$. □

Theorem 4.20. *Two ABA frameworks $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ and $D = (\mathcal{L}, \mathcal{R}', \mathcal{A}, ^-)$ are strongly equivalent w.r.t. stb semantics iff they have the same SE-Models.*

Proof. $(\Leftarrow)$ Given H we have to show that $D \cup H$ and $D' \cup H$ are equivalent.

Let X be conflict-free and a minimal model of $(D \cup H)^X$. Then X is a model of $D^X \cup H^X$ and hence a model of both D^X and H^X. From $X \models D^X$ it follows that (X, X) is an SE-model of D. By assumption, (X, X) is also an SE-model of D' and thus, X is a model of $(D')^X$. Thus, it is a model of $(D' \cup H)^X$ (as shown above, X is a model of H^X).

Suppose X is not minimal. Then there is some $Z \subsetneq X$ s.t. Z is a model of $(D' \cup H)^X$. Then Z is a model of $(D')^X$. From $Z \subseteq X$ and the two conditions $Z \models (D')^X$ and $X \vdash (D')^X$ it follows that (Z, X) is an SE-model of D'. Hence it is an SE-model of D by assumption and thus, Z is a model of $(D \cup H)^X$, contradicting the choice of X.

$(\Rightarrow)$ Suppose (X, Y) is an SE-model of D, but not of D'.

(case 1: Y is no model of $(D')^Y$). Let H be the ABA framework induced by the rules $\{y \leftarrow . \mid y \in Y \setminus \mathcal{A}\}$, i.e., $H = (Y, \mathcal{R}_H, \emptyset, \emptyset)$ with

$$\mathcal{R}_H = \{y \leftarrow . \mid y \in Y \setminus \mathcal{A}\}.$$

Since Y is no model of $(D')^Y$, it is also no model of the Y-reduct $(D' \cup H)^Y = (D')^Y \cup H^Y$. On the other hand, Y is a minimal model of D and hence also a minimal model of the Y-reduct $(D \cup H)^Y$, i.e., by applying Proposition 4.15 we find that H is our counter-example for strong equivalence of D and D'.

(case 2: Y is a model of $(D')^Y$).

Recall that due to Lemma 4.19, X and Y agree on their assumptions, i.e., if $p \in Y \setminus X$, then $p \notin \mathcal{A}$. Let H be the ABA framework induced by the rules

$$\mathcal{R}_H = \{x \leftarrow . \mid x \in X \setminus \mathcal{A}\} \cup \{p \leftarrow q \mid p, q \in Y \setminus X\},$$

i.e., $H = (X \setminus \mathcal{A} \cup Y \setminus X, \mathcal{R}_H, \emptyset, \emptyset)$. Since $X \subseteq Y$, we infer that Y is a model of H^Y and thus, a model of $(D' \cup H)^Y$ as well.

Suppose $Z \subseteq Y$ is a model of $(D' \cup H)^Y = (D')^Y \cup H^Y$. By choice of H, we must have $X \setminus \mathcal{A} \subseteq Z \setminus \mathcal{A}$; otherwise Z would not be a model of H^Y. Since (X, Y) is no SE-model of D', X is no model of $(D')^Y$ (the other conditions are met). Since X and Y agree on their assumptions, the reason must be some ordinary atom. Thus $X \setminus \mathcal{A} \neq Z \setminus \mathcal{A}$, i.e., there is some $p \in Y \setminus X$ with $p \in Z$, but $p \notin \mathcal{A}$. Hence, the rules "$p \leftarrow q$" with $p, q \in Y \setminus X$ in H are active in Z and thus, we must have $Y \setminus X \subseteq Z$ since otherwise Z would not be a model of H. We therefore have $Y \setminus \mathcal{A} \subseteq Z \setminus \mathcal{A}$. Moreover, $Y \cap \mathcal{A} \subseteq Z \cap \mathcal{A}$ must hold by definition of $(D' \cup H)^Y$, for otherwise Z would not be a model of the reduct. Consequently, $Y \subseteq Z$. Hence we deduce that Y is a minimal model of $(D' \cup H)^Y$.

On the other hand, $X \subsetneq Y$ s.t. X is a model of $(D \cup H)^Y$, i.e., Y is no minimal model of $(D \cup H)^Y = D^Y \cup H$ and H is the required counter-example. □

Example 4.21. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ and $D' = (\mathcal{L}, \mathcal{R}', \mathcal{A}, \overline{})$ be the ABA frameworks where $\mathcal{L} = \{a, b, p, \overline{a}, \overline{b}\}$, $\mathcal{A} = \{a, b\}$, and the rules are given as*

$\mathcal{R}:$	$\overline{a} \leftarrow b.$	$\overline{b} \leftarrow a.$	$\overline{b} \leftarrow p.$
$\mathcal{R}':$	$\overline{a} \leftarrow b.$	$\overline{b} \leftarrow a.$	

As adding a novel rule $p \leftarrow .$ would witness, D and D' are not strongly equivalent. Therefore, their SE models must differ. Indeed, consider (Y, Y) with $Y = \{\overline{a}, p\}$. This is no SE model of D since D^Y contains the rules

$$\mathcal{R}_{D^Y}: \qquad \overline{a} \leftarrow . \qquad \overline{b} \leftarrow p.$$

and hence $\overline{b} \in Th_{D^Y}(\emptyset)$, but $\overline{b} \notin Y$, i.e., $Y \not\models D^Y$. On the other hand, $(D')^Y$ contains only the rule

$$\mathcal{R}_{(D')^Y}: \qquad \overline{a} \leftarrow .$$

So $Y \models (D')^Y$ and (Y, Y) is an SE-model of D'. It is therefore rightfully spotted that D and D' are not strongly equivalent (and the SE model Y of D' also reflects the responsibility of the atom p).

Due to the close relation between our ABA fragment of consideration and LPs, it is not surprising that a characterization of strong equivalence of this kind can be found. The additional difficulty lies in the handling of the assumptions in our candidate set X, but for the most part, our proof technique follows (Turner, 2001).

5. An Abstract Instantiation Procedure for Dynamic Reasoning

In this section we will augment the standard ABA instantiation procedure with some additional information in order to make it better suitable for dynamic scenarios. Thereby, we will obtain so-called cvAFs ("claim and vulnerability augmented AFs") which extend AFs with additional information concerning the occurring arguments. It is, however, clear that an exact correspondence of the reasoning problems in ABA and cvAFs would again yield an intractable notion of enforcement and strong equivalence. This is why our cvAFs will be developed in a way that they carry just enough information in order to correspond to a meaningful fragment of ABA, while the aforementioned tasks stay tractable. This way, we obtain tractable fragments for Theorems 4.5 and 4.8.

Instantiated Arguments. Our cvAFs incorporate a crucial observation regarding the instantiations of knowledge bases which adhere well-formedness: arguments are typically characterized by their *claim* and their potential weaknesses (*vulnerabilites*) on which they can be attacked. In contrast, usual AF instantiations of knowledge bases disregard this information and output an abstract AF.

In ABA, the claim $cl(x)$ of an instantiated argument x is the conclusion of an argument (as expected) and the vulnerabilities $vul(x)$ correspond to the contraries of the assumptions used to derive the claim: for an argument $S \vdash p$ we have $cl(x) = p$ and $vul(x) = \overline{S}$.

However, note that this representation is not restricted to ABA. The vulnerabilities of arguments obtained from logic programs correspond to the negated atoms of the rules used in the construction. For logic-based argumentation, the vulnerabilities of an argument can correspond to the set of negated premises or even all formulas equivalent to it, also the conclusion of an argument can be a vulnerability (Gorogiannis & Hunter, 2011); for ASPIC$^+$, we furthermore consider the negation of defeasible rules as part of the vulnerabilities. Instantiated arguments thus provide a uniform representation for arguments with claims and defeasible elements.

Definition 5.1. *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. *For a set* X *of instantiated arguments we let* $cl(X) = \bigcup_{x \in X}\{cl(x)\}$.

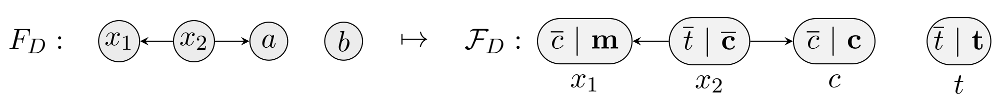
As mentioned above, in ABA frameworks we obtain instantiated arguments as follows:

for an ABA argument $S \vdash p$, we obtain the instantiated argument $(\overline{S}, p)$.

We are ready to formally introduce cvAFs as generalization of AFs by replacing abstract arguments with instantiated arguments.

Definition 5.2. *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 $\bar{c}$ and conclusion m.



cvAFs and AFs. When we remove claim and vulnerabilities of the cvAF and interpret the arguments as abstract entities we obtain a direct correspondence between cvAFs and usual Dung-style AFs. Hence our cvAFs are a proper generalization of AFs.

Notation 5.3. *For a cvAF $\mathcal{F} = (A, R)$, we let $F = (A, R)$ denote the corresponding AF.*

The basic concepts we introduced for AFs naturally transfer to cvAFs, simply because they can be viewed as AFs themselves (by ignoring the structure of the instantiated arguments). For example, the set E^+ of arguments attacked by $E \subseteq A$ in a cvAF $\mathcal{F} = (A, R)$ is $E^+ = \{a \in A \mid \exists e \in E : (e, a) \in R\}$. Analogously, the characteristic function of $\mathcal{F}$ is defined as $\Gamma_{\mathcal{F}}(E) = \{a \in A \mid E \text{ defends } a\}$. We also define the semantics of $\mathcal{F}$ by means of the underlying AF. This can be done in terms of arguments or claims.

Definition 5.4. *Given an cvAF $\mathcal{F}$ and an AF semantics σ. We let*

- $\sigma(\mathcal{F}) = \sigma(F)$ *denote the σ-argument-extensions and*
- $\sigma_{cl}(\mathcal{F}) = \{cl(E) \mid E \in \sigma(F)\}$ *denote the σ-conclusion-extensions of $\mathcal{F}$.*

Well-formed cvAFs. 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. We call a cvAF adhering to this property *well-formed.*

Definition 5.5. *A cvAF $\mathcal{F} = (A, R)$ is called* well-formed *if $(x, y) \in R$ iff $cl(x) \in vul(y)$ for each $x, y \in A$.*

Remark 5.6. *The term* well-formedness *has been coined in the context of claim-focused argumentation (Dvořák & Woltran, 2020) and formalizes that arguments with the same conclusion attack the same arguments. We note that a cvAF can be well-formed in the sense of (Dvořák & Woltran, 2020) without adhering to the requirement from Definition 5.5. In our formulation, we exclude situations in which arguments are attacked without having the claim of the attacker as vulnerability. Hence, the adaption of the concept of well-formedness to cvAFs as presented in Definition 5.5 is more restrictive; however, it takes the attack construction of the underlying knowledge base into account. With our adaption, we capture the intuition that an attack must be justified by the presence of vulnerabilities.*

cvAFs and ABA frameworks. Let us now see or formalism at work when applied to ABA frameworks. We adapt the standard instantiation (cf. Definition 2.5) as follows.

Definition 5.7. *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. By construction of the attack relation in ABA we obtain that each cvAF is well-formed.

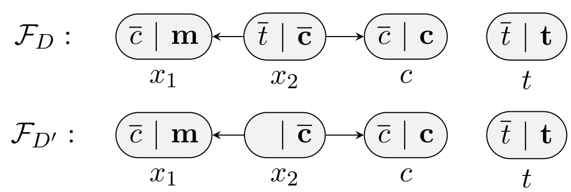
Proposition 5.8. *For each ABA D, its associated cvAF $\mathcal{F}_D$ is well-formed.*

Moreover, the instantiation preserves the semantics from the original instance.

Proposition 5.9. *Let D be an ABA framework and $\mathcal{F}_D$ its associated cvAF. For each $\sigma \in \{gr, co, pr, stb\}$, it holds that $\sigma_{Th}(D) = \sigma_{cl}(\mathcal{F}_D)$ and $\sigma(D) = \{S \cap \mathcal{A} \mid S \in \sigma_{cl}(\mathcal{F}_D)\}$.*

Proof. Since our cvAF instantiation generalizes the standard instantiation, the results also apply to our case. By Proposition 2.6, it holds that (i) if $E \in \sigma(\mathcal{F}_D)$ then $asms(E) \in \sigma(D)$; and (ii) if $S \in \sigma(D)$ then $\{S' \vdash p \mid \exists S' \subseteq S, R \subseteq \mathcal{R} : S' \vdash_R p\} \in \sigma(\mathcal{F}_D)$. By definition of Th_D, we obtain $\sigma_{Th}(D) = \sigma_{cl}(\mathcal{F}_D)$. Moreover, when restricting the extensions to the conclusions that correspond to assumptions, we obtain $\sigma(D) = \{S \cap \mathcal{A} \mid S \in \sigma_{cl}(\mathcal{F}_D)\}$. □

Example 5.10. *When instantiating our ABA frameworks D and D' from Example 1.1 as cvAFs, we obtain the following results:*



Comparing these with our AF instantiations from Example 1.1, we observe a crucial difference: while the AFs corresponding to D and D' are identical, the cvAFs $\mathcal{F}_D$ and $\mathcal{F}_{D'}$ differ: the argument x_2 has vulnerability $vul_D(x_2) = \{\bar{t}\}$ in $\mathcal{F}_D$ but no vulnerabilities in $\mathcal{F}_{D'}$.

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

Assumption 5.11. *In the remainder of this work, we assume that each cvAF is well-formed.*

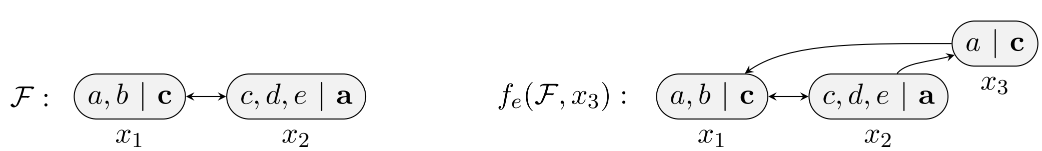
cvAFs and Dynamics. We are ready to investigate dynamics in structured argumentation by means of cvAFs. Suppose we are given a knowledge base $\mathcal{K}$ and the instantiated cvAF $\mathcal{F}_\mathcal{K}$. If we want to move to a superset $\mathcal{K} \cup \mathcal{H}$ we can construct $\mathcal{F}_{\mathcal{K}\cup\mathcal{H}}$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

Definition 5.12. *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 5.13. *Consider the cvAF $\mathcal{F}$ with mutually attacking arguments $x_1 = (\{a, b\}, c)$ and $x_2 = (\{c, d, e\}, a)$. The expansion of $\mathcal{F}$ with argument $x_3 = (\{a\}, c)$ induces the attacks (x_3, x_1) and (x_2, x_3). We depict both cvAFs below.*



cvAFs and Atomic ABA Frameworks. Our cvAFs are closely related to a certain class of ABA frameworks.

Definition 5.14. *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 several decisive observations we make about atomic ABA frameworks.

Lemma 5.15. *Given an atomic ABA* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, {}^{-})$ *and the corresponding cvAF* $\mathcal{F}_D$.

- *for each atomic rule* $r = p \leftarrow S$, *in* D *we have* $\mathcal{F}_{D \cup \{r\}} = f_e(\mathcal{F}_D, x)$ *with* $x = (\overline{S}, p)$;
- *for each* $x = (\overline{S}, p)$, *we have* $\mathcal{F}_{D \cup H} = f_e(\mathcal{F}_D, x)$ *with* $H = (\mathcal{L}, \{p \leftarrow S\}, S, {}^{-})$.

By moving from general to atomic ABA we do not lose expressive power; each framework can be transformed into an atomic one, which makes this fragment particularly interesting.

Proposition 5.16. *For each ABA* D *there is an atomic ABA* D' *such that* $\sigma(D) = \sigma(D')$ *for each semantics under consideration.*

Proof. The underlying idea is that for a given ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, {}^{-})$, we can construct a corresponding atomic ABA $D' = (\mathcal{L}, \mathcal{R}', \mathcal{A}, {}^{-})$ by iteratively replacing rules $r : p \leftarrow S$ that contain non-assumptions in their body, i.e., some $s \in S \setminus \mathcal{A}$, by rules $r' : p \leftarrow (S \setminus \{s\}) \cup body(s)$, until a fixed point is found.

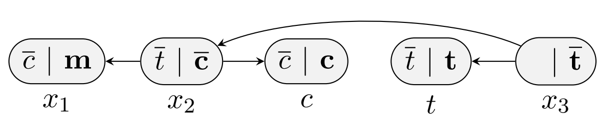
We prove this statement via a small detour over cvAF instantiations: First, instantiate D as cvAF $\mathcal{F}_D$ and mark all arguments that stem from non-assumptions (i.e., all arguments that are not of the form $(\overline{a}, a)$ for some $a \in \mathcal{A}$). Second, by Lemma 5.15, each argument $x = (\mathcal{S}, p)$ in $\mathcal{F}_D$ corresponds to an atomic ABA $H = (\mathcal{L}, \mathcal{R}'', \mathcal{A}'', {}^{-})$ with $R' = \{p \leftarrow S\}$ and $\mathcal{A}' = \overline{S}$. We extract our atomic ABA by transforming each argument that stems from a non-assumption to an atomic ABA H and form the union of all such ABAs to obtain D'. Lastly, we add potentially missing assumptions (assumptions that do not appear in the body of any rule) to our set of assumptions. Proposition 5.8 guarantees the semantical correpondence of D and D'. □

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

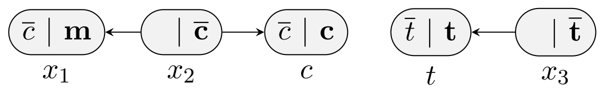
Example 5.18. *Let* D *be our running example and* $\mathcal{F}_D$ *its instantiated cvAF as depicted above. Adding a fact "*$\bar{t}$.*" yields an additional instantiated argument* $x_3 = (\emptyset, \bar{t})$: *Since* $\mathcal{F}_D$ *rightfully encodes that conclusion* $\bar{t}$ *is a threat to* x_2, *the instantiation of the resulting ABA framework can be directly computed from* $\mathcal{F}_D$ *by adding the argument* $x_3 = (\emptyset, \bar{t})$.

$f_e(\mathcal{F}_D, x_3)$:



If we consider instead the expansion of $\mathcal{F}_{D'}$ instantiated from the ABA framework D' from Example 1.1 with the same argument x_3, we obtain the following picture:

$f_e(\mathcal{F}_{D'}, x_3)$:



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.

6. Making the Enforcement Problem Tractable

In this section we develop a notion of the enforcement problem for cvAFs and establish criteria for deciding enforceability. At first glance, this yields results applicable to atomic ABAs due to Lemma 5.15; however, we will discuss some subtle details of the notions which one needs to be aware of.

We note that we do not restrict the space of possible expansions in our analysis. That is, we assume that each potential argument can be added to a given cvAF.

6.1 Claim and Argument Enforcement in cvAFs

In line with our enforcement notion from Definition 4.2, 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 6.1. *Let $\mathcal{F} = (A, R)$ be a cvAF and σ a semantics. We say that*

- *a conclusion p is σ-*enforceable *in $\mathcal{F}$ if there is a set X of instantiated arguments s.t. $p \notin cl(X)$ and p is credulously accepted in $f_e(\mathcal{F}, X)$;*
- *an argument $x \in A$ is σ-*enforceable *in $\mathcal{F}$ if there is a set X of instantiated arguments s.t. $cl(x) \notin cl(X)$ and x is credulously accepted in $f_e(\mathcal{F}, X)$.*

Example 6.2. *Let $\mathcal{F}_D$ be our running example cvAF and consider the expansion $f_e(\mathcal{F}_D, x_3)$ with $x_3 = (\emptyset, \overline{t})$ (cf. Example 5.18). Since $co(f_e(\mathcal{F}_D, x_3)) = \{\{c, x_1, x_3\}\}$ with $cl(x_1) = m$ we obtain that conclusion m is co-enforceable.*

In the following we establish criteria to decide whether arguments and conclusions are enforceable in cvAFs. By definition, it suffices to focus on argument enforcement:

Proposition 6.3. *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 5.18).

In general, arguments without any vulnerability will always be accepted in complete-based semantics; and thus, all arguments they attack will be defeated. This is not only the case within the given cvAF, but also for any conceivable expansion. Motivated by this observation, we call the affected arguments *strongly defeated.*

Definition 6.4. *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 $vul(y) = \emptyset$.*

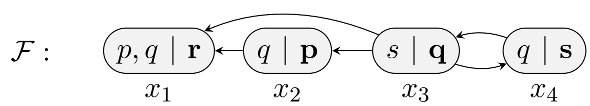
Example 6.5. *In our running example involving the cvAF that results from an instantiation of D', the argument x_1 is strongly defeated. In fact, it is verifiable with reasonable effort that x_2 is part of the grounded extension in any possible expansion $f_e(\mathcal{F}, X)$.*

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

Proposition 6.6. *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 6.7. *Consider the cvAF $\mathcal{F}$ depicted below.*



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 6.8. *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 $vul(y) \subseteq vul(x)$. In this case, we call y the* witness *for strong unacceptability of 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 6.9. *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.*

Proof. ($\Rightarrow$) Suppose x is strongly unacceptable and let $x \in A$ with $(y, x) \in R$ and $vul(y) \subseteq vul(x)$. Assume that in some expansion $f_e(\mathcal{F}, X)$ we have $x \in E$ for $E \in ad(f_e(\mathcal{F}, X))$. Since E defends x, E attacks y. Due to $vul(y) \subseteq vul(x)$, E attacks x as well; contradiction.

($\Leftarrow$) Suppose x is not strongly unacceptable. First consider $\sigma \neq stb$. Then x is no self-attacker since otherwise it would be strongly unacceptable: Let $y := x$ and we get $(y, x) \in R$ and $vul(y) \subseteq vul(x)$. So let $y_1, \ldots, y_n$ be the set of arguments in $\mathcal{F}$ attacking x which are not counter-attacked by x, i.e., we have $cl(x) \notin vul(y_i)$. Moreover, for each i we have $vul(y_i) \setminus vul(x) \neq \emptyset$ since x is not strongly unacceptable. Consider some $p_i \in$

$vul(y_i) \setminus vul(x) \neq \emptyset$ for each i. Let $x_i = (p_i, \emptyset)$ be unattacked instantiated arguments with the p_i as respective claim. Set $X = \{x_1, \ldots, x_n\}$. It is straightforward to see that $x \cup X$ is admissible in $f_e(\mathcal{F}, X)$. By choice of the y_i, $cl(x) \notin cl(X)$.

Now let $\sigma = stb$. Here in addition we need to ensure that there is at least one extension in our expansion. We will proceed by taking care of each self-attacker as well as each odd cycle in the given cvAF.

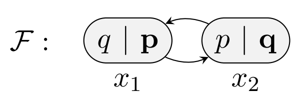
- Let $z_1, \ldots, z_m$ be the set of self-attacking arguments in $\mathcal{F}$ (not attacked by x). Suppose for some i we have $vul(z_i) \subseteq vul(x)$. Then since $(z, z) \in R$ we also have $(z, x) \in R$ and we infer strong unacceptability of x; contradiction. So we infer $vul(z_i) \nsubseteq vul(x)$ for each i. Therefore, we can find $q_i \in vul(z_i) \setminus vul(x) \neq \emptyset$ for each i. Let $x'_i = (q_i, \emptyset)$ be unattacked instantiated arguments with the q_i as respective claim. Set $X' = \{x'_1, \ldots, x'_n\}$; due to X' we have considered each self-attacking argument in $\mathcal{F}$.

- Let $O = \{o_1, \ldots, o_n\}$ be an arbitrary but fixed set of arguments forming an odd cycle in $\mathcal{F}$ (not attacked by x). We argue that there is some $o_i \in O$ satisfying the usual condition, i.e., $vul(o_i) \nsubseteq vul(x)$. Otherwise each $o_i \in O$ attacks x since $(o_i, o_j) \in R$ implies $(o_i, x) \in R$ due to $vul(o_j) \subseteq vul(x)$ (we get $(o_i, o_j) \in R$ from the fact that O is a cycle). Then x is strongly unacceptable; a contradiction. So take $o_i \in O$ with $vul(o_i) \nsubseteq vul(x)$ and construct an argument x'' attacking it as usual. Since O was arbitrary, we proceed like this for each odd cycle, obtaining a third set X'' of arguments.

In $f_e(\mathcal{F}, X \cup X' \cup X'')$ the set $x \cup X \cup X' \cup X''$ is admissible and each odd cycle is resolved. Due to (Baumann & Ulbricht, 2021, Theorem 5.7), $x \cup X \cup X' \cup X''$ can be extended to a stable extension of $\mathcal{F}$. □

For grounded semantics, however, we need to consider further unacceptability notions. The reason why Theorem 6.9 does not hold for grounded semantics is that an argument might be capable of defending itself, but still not be part of the iterative procedure which yields the grounded extension. To illustrate this we consider the following example.

Example 6.10. *Suppose we aim to gr-enforce x_1 in $\mathcal{F}$:*



Since $gr(\mathcal{F}) = \emptyset$ we would have to introduce an argument defeating x_2 in order defend x_1 from the incoming attack. However, such an argument has conclusion p 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 6.11. *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 $vul(y) \setminus \{cl(x)\} \subseteq vul(x)$. In this case, we call y the* witness *for strong gr-unacceptability of x.*

Indeed, this condition is violated in Example 6.10: We observe that in this case, we have $vul(x_2) \setminus cl(x_1) = \{p\} \setminus \{p\} = \emptyset$. We see that strong gr-unacceptability formalizes that the only conceivable attacker of y is x itself; any other conclusion attacking y would result in attacking x as well, similar in spirit to the strong unacceptability notion from above.

The following condition characterizes gr-enforceability for cvAFs. Unfortunately, we do not get a straightforward characterization of the form "enforceable iff not strongly gr-unacceptable". The intuitive reason is as follows. Even though x_1 is strongly gr-unacceptable in Example 6.10, it might be enforceable if another argument z with claim p was present: In this case, we could try to enforce y which could then in turn defend x_1 from x_2. For this to be possible, we would have to check our usual requirements for this conceivable argument y, i.e., can we defend y without also attacking x_1? The following proposition formalizes this intuition.

Proposition 6.12. *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 $cl(y) = cl(x) = q$ such that*
  - *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 $vul(z) \setminus (vul(x) \cup vul(y)) \neq \emptyset$.*

Proof. ($\Leftarrow$) First suppose x is not gr-unacceptable. As usual let $w_1, \ldots, w_n$ be the set of attackers of x. We have $(vul(w_i) \setminus \{cl(x)\}) \setminus vul(x) \neq \emptyset$, so we take one conclusion $p_i \in (vul(w_i) \setminus \{cl(x)\}) \setminus vul(x)$, introduce corresponding instantiated arguments $(p_i, \emptyset)$ and obtaining a set X s.t. x is defended by X in $f_e(\mathcal{F}, W)$.

Now suppose the second condition is true and consider $y \in A$ as described.

- Let $z_1, \ldots, z_n$ be the set of arguments attacking y. As usual, we take conclusions $p_i \in vul(z_i) \setminus (vul(x) \cup vul(y) \cup \{q\})$.
- Let $z'_1, \ldots, z'_m$ be the set of arguments attacking x. For each z'_i with $q \notin vul(z'_i)$ consider a conclusion $q_i \in vul(z) \setminus (vul(x) \cup vul(y))$.

Let Z be the set of instantiated arguments with the considered conclusions as claims and no vulnerabilities. By construction, Z defends y in $f_e(\mathcal{F}, Z)$ and defeats each attacker of x not having q as vulnerability; arguments of this kind are defeated due to y being defended. That is, $Z \cup \{x, y\}$ is part of the grounded extension of $f_e(\mathcal{F}, Z)$.

($\Rightarrow$) Suppose both conditions are false, i.e., x is strongly gr-unacceptable and there is no y satisfying the two mentioned conditions. If x is even strongly unacceptable, we are done since this would even prevent us from enforcing x w.r.t. co semantics.

So suppose x is not strongly unacceptable, but strongly gr-unacceptable. Then there is some argument z attacking x and $vul(z) \setminus \{q\} \subseteq vul(x)$. Hence, in order for x to be in the grounded extension, we need to ensure defense of some argument different from x with claim q. Take some y with $cl(y) = q$ (if none exists, we are done). By assumption, at least one of the mentioned conditions is wrong.

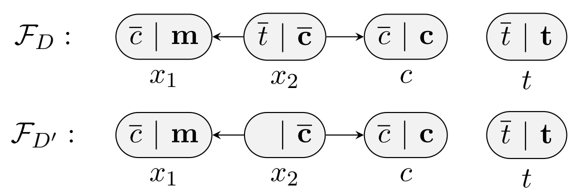
- Suppose $vul(z) \setminus (vul(x) \cup vul(y) \cup \{q\}) = \emptyset$ for some attacker z of y. However, this means by introducing an argument not having q as conclusion we can never ensure defeat of z without also defeating either x or y. We cannot introduce arguments which defeat x and defeating y means we need to move to another y' having claim q.

- Now suppose some z attacking x with $q \notin vul(z)$ satisfies $vul(z)\setminus(vul(x)\cup vul(y)) = \emptyset$. As before, this means we cannot defend x from z without introducing arguments which also defeat either x or y; again this means that we need to move to another y'. □

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 6.9 and Proposition 6.12 we obtain:

Corollary 6.13. *Let $\mathcal{F} = (A, R)$ be a cvAF and $\sigma \in \{ad, co, pr, stb\}$. A conclusion $p \in cl(A)$ is σ-enforceable if and only if there is an argument $x \in A$ with $cl(x) = p$ and x is not strongly unacceptable; it is gr-enforceable if and only if there is an argument $x \in A$ with $cl(x) = p$ and x is not strongly gr-unacceptable.*

Example 6.14. *Recall our introductory examples.*



In both $\mathcal{F}_D$ and $\mathcal{F}_{D'}$, x_2 and t are accepted on their own. Moreover, in $\mathcal{F}_D$ we have that

- *x_1 and c can be enforced by introducing an argument with claim $\bar{t}$;*

- *no argument is strongly gr-unacceptable.*

In $\mathcal{F}_{D'}$ we have that

- *x_1 and c are both strongly unacceptable and gr-unacceptable (even strongly defeated) and thus cannot be enforced.*

6.2 Enforcing Sets in cvAFs

In the previous subsection, our investigation was focusing on enforcement of a single claim resp. argument. Let us now go one step further and consider enforcing sets instead. Interestingly, the results we obtained so far do not generalize to this setting.

Definition 6.15. *Let $\mathcal{F} = (A, R)$ be a cvAF and σ a semantics. We say that*

- *a set C of conclusions is* σ-enforceable *if there is a set X of instantiated arguments s.t. $cl(X) \cap C = \emptyset$ and $C \subseteq E$ for some $E \in \sigma_{cl}(f_e(\mathcal{F}, X))$;*

- *a set $Y \subseteq A$ of arguments is* σ-enforceable *if there is a set X of instantiated arguments s.t. $cl(X) \cap cl(Y) = \emptyset$ and $Y \subseteq E$ for some $E \in \sigma(f_e(\mathcal{F}, X))$.*

Again, let us start with argument enforcement. Naturally, our strategy is to utilize our previously developed techniques in order to enforce all of them simultaneously. For this, recall strong unacceptability: If x is *not* strongly unacceptable, then for each attacker y of x we have

$$vul(y) \setminus vul(x) \neq \emptyset$$

which ensures that we can introduce a novel argument z with claim $p \in vul(y) \setminus vul(x)$ which defends x against y. Now suppose we want to enforce x' as well. In this case, we need to defend x against y while also not defeating x', i.e., we require that

$$vul(y) \setminus (vul(x) \cup vul(x')) \neq \emptyset$$

in order to proceed analogously.

We turn these observations into a general notion as follows.

Definition 6.16. *For a cvAF $\mathcal{F} = (A, R)$, $x \in A$ is C-*compatibly enforceable *if for each attacker y of x we have that $vul(y) \setminus C \neq \emptyset$.*

This notion generalizes strong unacceptability in the sense that x is not strongly unacceptable iff it is $vul(x)$-compatibly enforceable. We obtain the following characterization for enforcing sets of arguments. Observe that we do not yet make any statement about gr semantics. This will be done later.

Theorem 6.17. *Let $\mathcal{F} = (A, R)$ be a cvAF and suppose $\sigma \in \{co, pr, stb\}$. A set X of arguments is σ-enforceable if and only if each $x \in X$ is $vul(X)$-compatibly enforceable.*

Proof. The proof is analogously to the one for Theorem 6.9. Observe that the stronger condition of $vul(X)$-compatibility ensures that we can enforce each $x \in X$ without defeating the other arguments we seek to accept. □

In view of these results regarding argument enforcement, one might now anticipate the following procedure for claim enforcement:

- Given a set C of claims, consider a set of arguments X s.t. $cl(X) = C$;
- check in polynomial time whether X can be enforced.

While the second part is indeed a simple check according to Theorem 6.17, the first item involves searching for a suitable set X of arguments. This guess indeed renders the problem intractable. To this end consider the following reduction which will serve as the basis for our lower bound.

Reduction 6.18. *For a CNF formula φ with clauses $C = \{c_1, \dots, c_n\}$ over variables in $X = \{x_1, \dots, x_m\}$, we define the corresponding cvAF $\mathcal{F} = (A, R)$ with $A = C \cup \{v_1, \dots, v_m\} \cup \{\bar{v}_1, \dots, \bar{v}_m\}$ where*

$$\begin{aligned} &\forall c \in C : cl(c) = c && vul(c) = \{x_j \mid x_j \in c\} \cup \{\bar{x}_j \mid \neg x_j \in c\} \\ &\forall 1 \leq i \leq m : cl(v_i) = T_i && vul(v_i) = C \cup \{\bar{x}_i\} \\ &\forall 1 \leq i \leq m : cl(\bar{v}_i) = T_i && vul(\bar{v}_i) = C \cup \{x_i\} \end{aligned}$$

and the induced attack relation. An example of this reduction can be found in Figure 2.

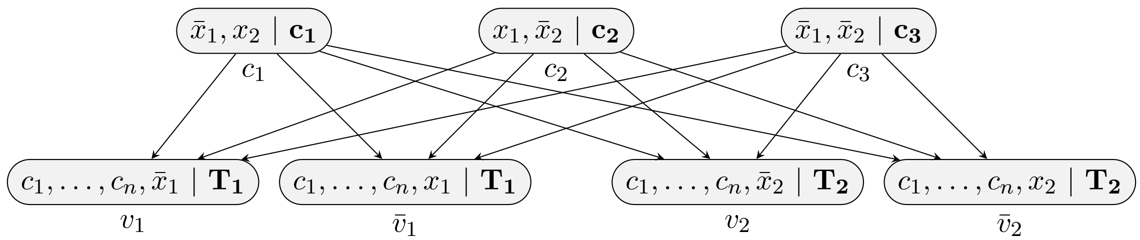


Figure 2: Reduction 6.18 applied to the formula ϕ consisting of clauses $c_1 = \{\neg x_1, x_2\}$, $c_2 = \{x_1, \neg x_2\}$, $c_3 = \{\neg x_1, \neg x_2\}$

Theorem 6.19. *Deciding whether a set C of claims is σ-enforceable for a given cvAF $\mathcal{F} = (A, R)$ and semantics $\sigma \in \{co, gr, pr, stb\}$ is* NP*-hard.*

Proof. Given a formula ϕ in CNF we construct $\mathcal{F} = (A, R)$ as described in Reduction 6.18. We claim that ϕ is satisfiable iff $\{T_1, \ldots, T_m\}$ can be enforced in $\mathcal{F}$.

($\Rightarrow$) Suppose $\omega : X \to \{0, 1\}$ is a satisfying assignment for ϕ. Let

$$W = \{(\emptyset, x_i) \mid \omega(x_i) = 1\} \cup \{(\emptyset, \bar{x}_i) \mid \omega(x_i) = 0\}$$

be a set of m instantiated arguments. Consider the expansion $f_e(\mathcal{F}, W)$. Since the vulnerabilities of the c_i arguments correspond to satisfying literals, each c_i is defeated in $f_e(\mathcal{F}, W)$. Moreover, since W corresponds to a satisfying assignment, for each $1 \le i \le m$ we have that either v_i or $\bar{v}_i$ is not attacked by W. Hence, $cl(W) \cup \{T_1, \ldots, T_m\} \in \sigma_{cl}(f_e(\mathcal{F}, W))$.

($\Leftarrow$) Let W be s.t. $\{T_1, \ldots, T_m\}$ is enforced in $f_e(\mathcal{F}, W)$. Then, for each $1 \le i \le n$, we have $x_i \notin cl(W)$ or $\bar{x}_i \notin cl(W)$ (or both), i.e., $cl(W) \cap (X \cup \bar{X})$ corresponds to some (partial) assignment. Since W must defend the v_i arguments against $c_1, \ldots, c_n$, $cl(W) \cap (X \cup \bar{X})$ must even correspond to a satisfying assignment. We therefore let

$$\omega : X \to \{0, 1\} \qquad \omega(x_i) = 1 \Leftrightarrow x_i \in cl(W) \qquad \omega(x_i) = 0 \Leftrightarrow x_i \notin cl(W)$$

and by this obtain a satisfying assignment ω for ϕ. □

We are left to discuss grounded semantics. Indeed, in this case argument enforcement is already intractable which can be shown by a suitable adjustment to Reduction 6.18. The intuitive reason why this problem is intractable can be found in the proof of Proposition 6.12: Recall that in case we cannot defend x directly, we might have to move to some other argument y having the same claim and then try to defend y additionally. So the procedure involves guessing a suitable candidate y. By moving to enforcement of sets of arguments, we would consequently have to guess a set of suitable candidates Y first. Proof details can be found in Appendix B.

Theorem 6.20. *Deciding whether a set X of arguments is gr-enforceable for a given cvAF $\mathcal{F} = (A, R)$ is* NP*-hard.*

6.3 Constrained Enforcement

So far, our enforcement notions were guided by the idea that acceptance of a certain claim shall be ensured and therefore it makes sense to stipulate that the target claim itself cannot be introduced. However, from a mere technical point of view, one could forbid any set of claims, a notion which we consider next. This will, however, turn out to be NP-hard in any case. While this is bad news on its own, this insight provides important inspirations for the later Section 6.4 where we find our tractable ABA fragment.

Let us delve into the technicalities.

Definition 6.21. *Let $\mathcal{F} = (A, R)$ be a cvAF and σ a semantics. We say that*

- *a set C of conclusions is D-eluding σ-*enforceable *if there is a set X of instantiated arguments s.t. $cl(X) \cap D = \emptyset$ and $C \subseteq E$ for some $E \in \sigma_{cl}(f_e(\mathcal{F}, X))$;*
- *a set $Y \subseteq A$ of arguments is D-eluding σ-*enforceable *if there is a set X of instantiated arguments s.t. $cl(X) \cap D = \emptyset$ and $Y \subseteq E$ for some $E \in \sigma(f_e(\mathcal{F}, X))$.*

It follows immediately by definition that D-eluding σ-enforceability coincides with the vanilla enforcement notion if we let $D = cl(X)$; we therefore face a faithful generalization of the previous version. This observation immediately yields the following lower bounds.

Corollary 6.22. *The following problems are* NP*-hard:*

- *deciding whether a set X of arguments is D-eluding gr-enforceable for a given cvAF $\mathcal{F} = (A, R)$;*
- *deciding whether a set C of claims is D-eluding σ-enforceable for a given cvAF $\mathcal{F} = (A, R)$ and semantics $\sigma \in \{co, gr, pr, stb\}$.*

However, this also holds for the remaining cases and hence, this notion is intractable for any semantics for both claim and argument enforcement. We can prove this by simply using the AF complexity standard reduction (Dvořák & Dunne, 2018, Reduction 3.6) and choosing D in a way that no relevant conclusion can be added. The proof can be found in Appendix B.

Theorem 6.23. *Deciding whether a set X of arguments (a set C of claims) is D-eluding σ-enforceable for a given cvAF $\mathcal{F} = (A, R)$ and semantics $\sigma \in \{co, gr, pr, stb\}$ is* NP*-hard.*

6.4 Consequences for Assumption-based Argumentation

Let us now head back to ABA. Recall the close correspondence between expansions within an atomic ABA framework D and expansions of the corresponding cvAF $\mathcal{F}_D$ as formalized in Lemma 5.15: Adding an atomic rule to D can be captured by a natural expansion of $\mathcal{F}_D$ and vice versa. Hence we would now hope to be able to transfer the tractable enforcement cases to atomic ABA as well. However, there is still a subtle issue which we did not yet take into consideration. First of all, although somewhat unexpected given the cvAF results, any enforcement notion we consider is intractable also for atomic ABAs. Indeed, when inspecting the construction for the proof of Theorem 4.5 we see that the constructed ABA framework is itself atomic.

Corollary 6.24. *Deciding whether assumption a (conclusion p) is enforceable w.r.t.* $\sigma \in \{co, gr, pr, stb\}$ *is* NP*-hard even for atomic ABA frameworks.*

So, how is this no contradiction to tractability in cvAFs? The reason is that we always assume our ABA frameworks to be *flat*, that is, we are not allowed to augment some framework D with a rule of the form $a \leftarrow .$ for some assumption $a \in \mathcal{A}$. However, for cvAFs we do not take this into consideration: Any set X of instantiated arguments induced a valid expansion $f_e(\mathcal{F}, X)$. Thus, it might happen that the resulting cvAF does not correspond to a flat ABA framework anymore due to $cl(X) \cap \mathcal{A} \neq \emptyset$. From a formal perspective this means there is a set of conclusions (namely the assumptions $\mathcal{A}$) which we are not allowed to introduce. Thus, translating the cvAF results to ABA would actually yield $\mathcal{A}$-eluding enforceability which we proved to be intractable in the previous subsection.

So, where is the tractable ABA fragment? 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 outgoing attacks.

Definition 6.25. *Let* $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ *be an ABA framework. We say D has* separated contraries *if* $\mathcal{A} \cap \overline{\mathcal{A}} = \emptyset$*.*

Now, it follows from the way our enforcement results are derived that we never rely on introducing arguments without any outgoing attacks. Hence, if D has separated contraries, we can apply our cvAF results while being certain that our expansions correspond to flat ABA frameworks, i.e., $cl(X) \cap \mathcal{A} \neq \emptyset$ always holds for any necessary $f_e(\mathcal{F}, X)$.

Theorem 6.26. *Deciding whether an argument or conclusion is enforceable w.r.t.* $\sigma \in \{co, gr, pr, stb\}$ *for atomic ABA frameworks with separated contraries is tractable.*

Proof. Let $(\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an atomic ABA framework with separated contraries. Recall that by Assumption 2.3, D is flat. We apply Lemma 5.15; from Proposition 2.6 formalizing the semantics correspondence we can infer that for each $p \in \mathcal{L}$ we have that p is enforceable in D iff the conclusion p is enforceable in $\mathcal{F}_D = (A_D, R_D)$ disregarding any expansion $f_e(\mathcal{F}, X)$ where $cl(X) \cap \mathcal{A} \neq \emptyset$. The proofs given for the enforcement results for cvAFs only require addition of arguments with outgoing attacks. Since D has separated contraries, $vul(A_D) \cap \mathcal{A} = \emptyset$ and we can assume $cl(X) \cap \mathcal{A} \neq \emptyset$ in each expansion $f_e(\mathcal{F}, X)$ without loss of generality, i.e., the conclusion p is enforceable in $\mathcal{F}_D = (A_D, R_D)$ disregarding any expansion $f_e(\mathcal{F}, X)$ where $cl(X) \cap \mathcal{A} \neq \emptyset$ iff the conclusion p is enforceable in $\mathcal{F}_D = (A_D, R_D)$. We hence infer p is enforceable in D iff the conclusion p is enforceable in $\mathcal{F}_D = (A_D, R_D)$. Now the claim follows since constructing $\mathcal{F}_D$ as well as the basic enforcement problem (Corollary 6.24) is tractable for atomic ABA frameworks. □

We also want to mention that by analogously applying Theorem 6.17 we make the same observation for enforcing sets of arguments (for $\sigma \neq gr$).

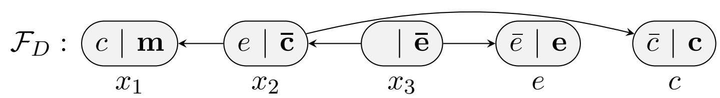
Theorem 6.27. *Deciding whether a set X of arguments is enforceable for atomic ABAs with separated contraries is tractable for* $\sigma \in \{co, pr, stb\}$*.*

We want to emphasize that moving from ABA to atomic ABA does not change the complexity class of the basic enforcement problem from Definition 6.1; but additionally requiring separated contraries does, i.e., we found a rather minor condition pushing the enforcement problem over the edge to tractability.

Example 6.28. *Recall ABA framework* $D = \{\mathcal{L}, \mathcal{A}, \mathcal{R}, \overline{\ \ }\}$ *from the discussion between Jane and Antoine, with the occurring atoms cinema* (c), $\overline{cinema}$ ($\bar{c}$), *expensive* (e), $\overline{expensive}$ ($\bar{e}$), *and movie* (m)*:*

$$\mathcal{L} = \{c,\ \bar{c},\ e,\ \bar{e},\ m\} \qquad \mathcal{A} = \{c,\ e\} \qquad \mathcal{R} = \{m \leftarrow c.,\quad \bar{c} \leftarrow e.,\quad \bar{e} \leftarrow .\}.$$

If we instantiate the corresponding cvAF $\mathcal{F}_D$ *we get the following graph.*



We now see that x_2 *is strongly defeated and thus infer that Antoine's argument* x_2 *is impossible to enforce.*

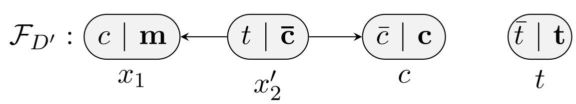
Recall that Antoine could also use the following argument x_2' *instead:*

> x_2' : *"If I do not like the trailer of this movie, I would prefer not to go to the cinema."*

Recall $D' = \{\mathcal{L}', \mathcal{A}', \mathcal{R}', \overline{\ \ }\}$ *where* t *is the abbreviation for Antoine not liking the trailer:*

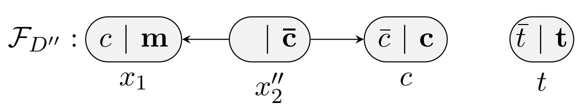
$$\mathcal{L}' = \{c,\ \bar{c},\ m,\ t,\ \bar{t}\} \qquad \mathcal{A}' = \{c,\ t\} \qquad \mathcal{R}' = \{m \leftarrow c.,\quad \bar{c} \leftarrow t.\}.$$

This time, we obtain:



Indeed, we see that now Antoine's argument is accepted, but Jane can enforce x_1 *by convincing Antoine that the trailer is nice (bringing forward some argument for* $\bar{t}$*).*

Recall the third variant where Antoine just says he doesn't want to go to the cinema without any condition. This would yield



with strongly defeated x_1 *and* c*, i.e., they cannot be enforced.*

We therefore see that the results we obtain from modeling the discussion as ABA framework and applying the cvAF results match our intuition.

Remark 6.29. *As a final remark in this section, let us mention that we do not lose any expressive power when insisting on separated contraries; if* $\overline{b} = a$*, then we can always add a fresh atom* p *as well as the rule* $p \leftarrow a.$ *and let* $\overline{b} = p$ *instead. After this little modification,* a *is not a contrary anymore. We note that this modification preserves the semantics considered in this work under projection to the set of original literals. Combined with Proposition 5.16 this yields that atomic ABA frameworks with separated contraries have the same expressiveness as ABA in general.*

7. Making Strong Equivalence Tractable

In this section, we establish methods to decide strong equivalence for two given cvAFs $\mathcal{F}$ and $\mathcal{G}$. 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 (Oikarinen & Woltran, 2011), Argumentation frameworks with collective attacks (SETAFs) (Dvořák, Rapberger, & Woltran, 2019), and claim-augmented argumentation frameworks (CAFs) (Baumann et al., 2022) 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 7.1. *Two cvAFs $\mathcal{F}$, $\mathcal{G}$ are* strongly equivalent *w.r.t. a semantics σ, denoted $\mathcal{F} \equiv_s^\sigma \mathcal{G}$, if for each set X of instantiated arguments $\sigma_{cl}(f_e(\mathcal{F}, X)) = \sigma_{cl}(f_e(\mathcal{G}, X))$ holds.*

Example 7.2. *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, if we recall the expansions of $\mathcal{F}_D$ and $\mathcal{F}_{D'}$ from Example 5.18 where we add the argument $x_3 = (\emptyset, \overline{b})$ to both frameworks, we obtain that $\{a, p, \overline{b}\}$ is stable in $f_e(\mathcal{F}_D, x_3)$ but not in $f_e(\mathcal{F}_{D'}, x_3)$. Hence $\mathcal{F}_D$ and $\mathcal{F}_{D'}$ are not strongly equivalent w.r.t. stable semantics.

In the above example, it was quite easy to come up with an appropriate counterexample. Not only that finding a counterexample 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, for each semantics we identify a specific kernel – checking strong equivalence then reduces to computing and comparing the respective kernels.

7.1 Redundancies

Let us start with some general observations regarding *redundancies* of cvAFs. For this, we first recall a redundancy notion which is also mentioned in the context of claim-augmented argumentation frameworks (CAFs) (Dvořák & Woltran, 2020). Here redundant arguments are identified and it is shown that we can safely delete them without changing (some) semantics. An argument x in a CAF $\mathcal{F}$ is called redundant w.r.t. argument y iff they have the same claim and attack the same arguments, but x is attacked by strictly more arguments than y, i.e., $y^- \subsetneq x^-$. This concept is naturally adapted to cvAFs as follows:

Definition 7.3. *For a cvAF $\mathcal{F} = (A, R)$ and argument $x \in A$ is* redundant *if there is $y \in A$ with $cl(y) = cl(x)$ and $vul(y) \subsetneq vul(x)$. In this case, y is called the* witness *for redundancy of x.*

Example 7.4. *The argument x_2 from the cvAF $\mathcal{F}_D$ from our running example is redundant w.r.t. $x = (\emptyset, \overline{a})$ because $cl(x) = cl(x_2) = \overline{a}$ and $vul(x) = \emptyset \subsetneq \{\overline{b}\} = vul(x_2)$.*

As shown in the literature (Dvořák, Rapberger, & Woltran, 2020), redundant arguments can be removed without changing the conclusion-σ-extensions of a given CAF for $\sigma \in \{gr, co, pr, stb\}$. This immediately translates to cvAFs.

Proposition 7.5. *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 $\sigma_{cl}(\mathcal{F}) = \sigma_{cl}(\mathcal{F} \setminus \{x\})$.*

Next, we reconsider the unacceptability notions from Section 6. We have shown that strongly defeated arguments cannot be enforced; in fact, they can be removed without changing the σ-extensions.

Proposition 7.6. *For a cvAF $\mathcal{F} = (A, R)$, semantics $\sigma \in \{gr, co, pr, stb\}$, and a strongly defeated argument $x \in A$, it holds that $\sigma_{cl}(\mathcal{F}) = \sigma_{cl}(\mathcal{F} \setminus \{x\})$.*

Proof. Let $\mathcal{F}' = \mathcal{F} \setminus \{x\}$, and let y with $vul(y) = \emptyset$ denote some argument which strongly defeats x. Observe that y is contained in the grounded extension of both $\mathcal{F}$ and $\mathcal{F}'$. It is easy to see that the grounded extension of $\mathcal{F}$ and $\mathcal{F}'$ coincide since $y \in \Gamma_{\mathcal{F}}(\emptyset)$ defeats x. Therefore,

$$\Gamma^i_{\mathcal{F}}(\emptyset) \subseteq \Gamma^i_{\mathcal{F}'}(\emptyset) \qquad \text{and} \qquad \Gamma^i_{\mathcal{F}'}(\emptyset) \subseteq \Gamma^{i+1}_{\mathcal{F}}(\emptyset).$$

We obtain

$$gr(\mathcal{F}) = \bigcup_{i \in \mathbb{N}} \Gamma^i_{\mathcal{F}}(\emptyset) = \bigcup_{i \in \mathbb{N}} \Gamma^i_{\mathcal{F}'}(\emptyset) = gr(\mathcal{F}').$$

Moreover, $E^+_{\mathcal{F}} \setminus \{x\} = E^+_{\mathcal{F}'}$ for every set of arguments E which is a superset of the (coinciding) grounded extension G. Hence $\Gamma_{\mathcal{F}}(E) = \Gamma_{\mathcal{F}'}(E)$ for each set $G \subseteq E$. Since each complete extension is a superset of G, we obtain $co(\mathcal{F}) = co(\mathcal{F}')$. It follows also that preferred semantics coincide. Regarding stable semantics, we argue analogously since each stable extension is a superset of G. □

For stable semantics we can make an even stronger assertion: Not only strongly defeated, but also strongly unacceptable arguments can be deleted without affecting the outcome.

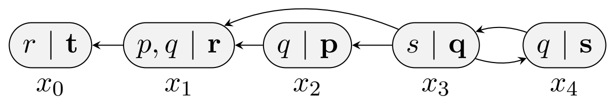
Proposition 7.7. *For a cvAF $\mathcal{F} = (A, R)$ and a strongly unacceptable argument $x \in A$, it holds that $stb_{cl}(\mathcal{F}) = stb_{cl}(\mathcal{F} \setminus \{x\})$.*

Proof. Let $\mathcal{F}' = \mathcal{F} \setminus \{x\}$ and let x be strongly unacceptable w.r.t. $y \in A$, i.e., $cl(x) = cl(y)$ and $vul(y) \subseteq vul(x)$. Observe that $E \in cf(\mathcal{F})$ iff $E \in cf(\mathcal{F}')$ for every E with $x \notin E$; moreover, x does not belong to any admissible extension of $\mathcal{F}$ and $\mathcal{F}'$, because if E defends x, then E defeats y and due to $vul(y) \subset vul(x)$, E defeats x as well. We obtain that x is either attacked by an admissible set or undecided. If y is contained in a stable extension, x is defeated; in case y is not contained in a stable extension, y is attacked and thus also x is attacked using $vul(y) \subseteq vul(x)$. Consequently, the argument x can be safely removed without changing the stable extensions of $\mathcal{F}$. □

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 7.8. *Consider cvAF $\mathcal{F}$ from Example 6.7 and a new argument $x_0 = (\{r\}, t)$:*

$f_e(\mathcal{F}, x_0):$



The resulting cvAF $f_e(\mathcal{F}, x_0)$ 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 render x_0 unattacked and thus change the grounded extension to $\{t\}$.

Strongly unacceptable arguments can neither be enforced nor deleted in such situations. This means that on the level of semantics, it is not possible to distinguish if such arguments are self-attacking or not. We show this by proving that the semantics of the cvAF remain unchanged after turning x into a self-attacker. Formally, this is achieved by removing it and expanding the resulting cvAF with some argument x' which is analogously to x, except having also its claim as vulnerability; formally, $x' = (vul(x) \cup \{cl(x)\}, cl(x))$.

Proposition 7.9. *For a cvAF $\mathcal{F} = (A, R)$, a semantics $\sigma \in \{gr, co, pr, stb\}$, and a strongly unacceptable argument $x \in A$, it holds that $\sigma_{cl}(\mathcal{F}) = \sigma_{cl}(f_e(\mathcal{F} \setminus \{x\}, x'))$ for $x' = (vul(x) \cup \{cl(x)\}, cl(x))$.*

Proof. Let $\mathcal{F}' = f_e(\mathcal{F} \setminus \{x\}, x')$ and assume x is strongly unacceptable w.r.t. $y \in A$. As outlined in the proof of Proposition 7.7, x can never appear in an admissible extension. We moreover observe that $E^+_F = E^+_{F'}$ for every conflict-free set E, $y \notin E$, since (x, x) is the only attack which has been introduced. We thus obtain $ad_{cl}(\mathcal{F}) = ad_{cl}(\mathcal{F}')$. Moreover, the grounded extension is preserved by adding this self-attack since it does not remove nor introduce new unattacked arguments (or any arguments defended by them). We thus obtain $\sigma_{cl}(\mathcal{F}) = \sigma_{cl}(\mathcal{F}')$ for $\sigma \in \{co, gr, pr, stb\}$. □

7.2 Complete Kernel for cvAFs

High level point of view. Strong equivalence for complete semantics can be characterized by comparing the *complete kernels* we define in this section, i.e., $\mathcal{F} \equiv^{co}_s \mathcal{G}$ iff their complete kernels coincide (Theorem 7.20). The complete kernel can be computed by the following procedure: Given $\mathcal{F}$,

1. turn each strongly unacceptable argument x into a self-attacker (i.e., formally add $cl(x)$ to the vulnerabilities $vul(x)$),

2. from the resulting cvAF remove all strongly defeated as well as redundant arguments.

This procedure yields the so-called *complete kernel* $\mathcal{F}^{ck}$. The main Theorem 7.20 of this section states that $\mathcal{F} \equiv^{co}_s \mathcal{G}$ iff $\mathcal{F}^{ck} = \mathcal{G}^{ck}$.

Technical details. Let us now formally introduce our first kernel. Following Proposition 7.9, the first adjustment we carry out is a modification on vulnerability level: Each strongly unacceptable argument x is turned into a self-attacker by adding $cl(x)$ to $vul(x)$. In the next step, we remove all strongly defeated and redundant arguments.

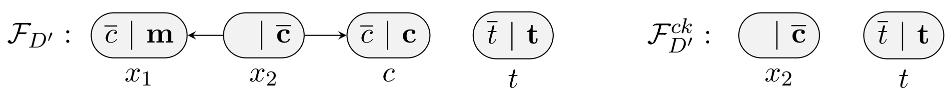
Definition 7.10. *For a cvAF $\mathcal{F} = (A, R)$, let X denote the set of all strongly unacceptable arguments in A and let*

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

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

$$\begin{aligned} A^{ck} &= A' \setminus \{x \in A' \mid x \text{ is strongly defeated or redundant}\}, \\ R^{ck} &= R' \cap (A^{ck} \times A^{ck}). \end{aligned}$$

Example 7.11. *The cvAF $\mathcal{F}_D$ from our running example coincides with its complete kernel since no arguments are strongly defeated, unacceptable or redundant. That is, we obtain $\mathcal{F}_D^{ck} = \mathcal{F}_D$. For $\mathcal{F}_{D'}$, we obtain the following picture:*



The goal of the following considerations is to show that the complete kernel characterizes strong equivalence for *co* semantics, i.e., $\mathcal{F} \equiv_s^{co} \mathcal{G}$ iff $\mathcal{F}^{ck} = \mathcal{G}^{ck}$. Our first step is to show that each cvAF is strongly equivalent to its kernel.

Proposition 7.12. *$\mathcal{F} \equiv_s^{\sigma} \mathcal{F}^{ck}$ for every cvAF $\mathcal{F}$ and for $\sigma \in \{co, gr, pr, stb\}$.*

Proof. Consider a set X of instantiated arguments. First, by Proposition 7.9, we can modify all strongly unacceptable arguments of $\mathcal{F}$ without changing semantics. Let $A_{unac} \subseteq A$ denote the set of unacceptable arguments in $\mathcal{F}$. For

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

we obtain $\sigma_{cl}(f_e(\mathcal{F}', X)) = \sigma_{cl}(f_e(\mathcal{F}, X))$. By Propositions 7.5 and 7.6, we can delete redundant and strongly unacceptable arguments as well. Let $A_{red} \subseteq A'$ and $A_{sdef} \subseteq A'$ denote the set of redundant and strongly defeated arguments of $\mathcal{F}'$, respectively. Then for

$$\mathcal{F}'' = \mathcal{F}' \setminus (A_{red} \cup A_{sdef})$$

we obtain $\sigma_{cl}(f_e(\mathcal{F}'', X)) = \sigma_{cl}(f_e(\mathcal{F}, X))$. By definition of the complete kernel, it holds that $\mathcal{F}'' = \mathcal{F}^{ck}$. We obtain $\sigma_{cl}(f_e(\mathcal{F}^{ck}, X)) = \sigma_{cl}(f_e(\mathcal{F}, X))$, hence $\mathcal{F} \equiv_s^{co} \mathcal{F}^{ck}$. □

In particular, this implies that the complete kernel of a cvAF preserves its semantics. This follows from the previous result by simply considering the empty expansion.

Corollary 7.13. *$\sigma_{cl}(\mathcal{F}) = \sigma_{cl}(\mathcal{F}^{ck})$ for every cvAF $\mathcal{F}$ and for $\sigma \in \{co, gr, pr, stb\}$.*

Next we show that kernelization behaves as expected: the complete kernel does not contain redundant and strongly defeated arguments; and each strongly unacceptable argument is self-attacking. For this, we consider the syntactical effects of our modifications. Let us start with the following fundamental observation.

Observation 7.14. *Removing arguments from a given cvAF $\mathcal{F}$ does not add novel redundant, strongly unacceptable, or strongly defeated arguments.*

Next we show that the modification of unacceptable arguments can be done iteratively.

Lemma 7.15. *Given a cvAF $\mathcal{F} = (A, R)$ and a strongly unacceptable argument $x \in A$. Let $x' = (vul(x) \cup \{cl(x)\}, cl(x))$ and let $\mathcal{F}' = f_e(\mathcal{F} \setminus \{x\}, x') = (A', R')$. Then, for all $y \neq x \in A$, y is strongly unacceptable in $\mathcal{F}$ iff y is strongly unacceptable in $\mathcal{F}'$.*

Proof. Consider a strongly unacceptable argument $y \in A$ in $\mathcal{F}$. Then there is $z \in A$ with $vul(z) \subseteq vul(y)$ and $(z, y) \in R$ in $\mathcal{F}$. First assume $z \neq x$. Then it holds that $z \in A'$, witnessing unacceptability of y in $\mathcal{F}'$. Otherwise, in case $z = x$, there is $z' \in A$ with $vul(z') \subseteq vul(x) = vul(z)$ such that $(z', x) \in R$. Consequently, $(z', y) \in R$ with $vul(z) \subseteq vul(y)$, showing that y is strongly unacceptable in $\mathcal{F}'$.

For the other direction, consider a strongly unacceptable argument $y \in A'$ in $\mathcal{F}'$. It holds that there is a witness $z \in A'$ of the strong unacceptability of y in $\mathcal{F}'$. By construction, z is also a witness in $\mathcal{F}$. □

We observe that we might obtain novel redundant arguments when turning unacceptable arguments into self-attackers.

Example 7.16. *Consider three arguments x, y, z with claims $cl(x) = cl(y) = c$ and $cl(z) = d$ and vulnerabilities $vul(x) = \{d, e, f\}$, $vul(y) = \{c, d, e\}$, and $vul(z) = \{e\}$. The arguments x and y are strongly unacceptable w.r.t. z because z attacks both of them and $vul(z)$ is a subset of both $vul(x)$ and $vul(y)$. The argument y is already a self-attacker. Turning x into a self-attacker yields the modified argument $x' = (\{c, d, e, f\}, c)$ which is redundant w.r.t. y.*

We show that redundant and strongly defeated arguments can be removed iteratively.

Lemma 7.17. *Given a cvAF $\mathcal{F}$ and arguments $x, y \in A$, $x \neq y$. Let y be redundant or strongly defeated in $\mathcal{F}$. Then x is redundant or strongly defeated in $\mathcal{F}$ iff x is redundant or strongly defeated in $\mathcal{F} \setminus \{y\}$.*

Proof. In case x is redundant or strongly defeated in $\mathcal{F} \setminus \{y\}$ then there is a witness z in $\mathcal{F} \setminus \{y\}$. As mentioned in Observation 7.14, the claim-attacks are not affected by removing certain arguments. We thus obtain that z witnesses that x is redundant or strongly defeated in $\mathcal{F}$. Also, in case x is strongly defeated in $\mathcal{F}$, it is clear that x is contained in $\mathcal{F} \setminus \{y\}$ since y cannot serve as witness of x being strongly defeated since $vul(y) \neq \emptyset$.

Now, let y be strongly defeated in $\mathcal{F}$. In case x is redundant w.r.t. y in $\mathcal{F}$, there is some $z \in A$ with $(z, y) \in R$. We obtain x is strongly defeated (using $vul(y) \subseteq vul(x)$).

Let y be redundant in $\mathcal{F}$ and let x be redundant w.r.t. y in $\mathcal{F}$. Then there is $z \in A$ with $vul(z) \subseteq vul(y)$ and $cl(z) = cl(y)$, thus witnessing the redundancy of x. □

We remark that the disjunction of the properties is preserved. That is, a redundant argument can turn into a strongly defeated argument when removing y.

Combining the previous Lemmata, we can be sure that the complete kernel is capable of taking care of all redundant, strongly defeated, and strongly unacceptable arguments. Formally, we make the following observation.

Proposition 7.18. *The complete kernel $\mathcal{F}^{ck}$ of a cvAF $\mathcal{F}$ does not contain redundant nor strongly defeated arguments, and each strongly unacceptable argument is self-attacking.*

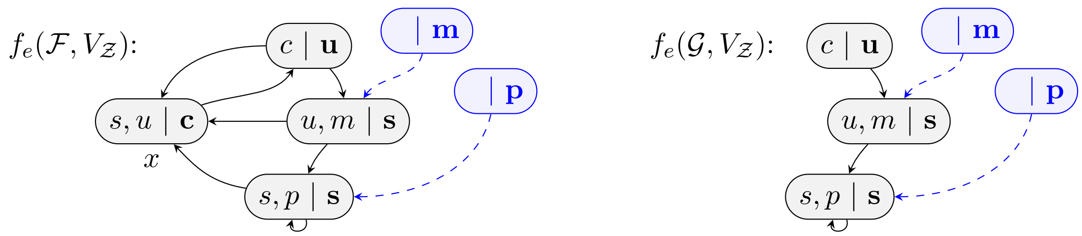


Figure 3: Illustration of Case 1 in the proof of Lemma 7.19 with $V_\mathcal{Z} = \{(\emptyset, m), (\emptyset, p)\}$.

Proof. We first modify strongly unacceptable arguments. By Lemma 7.15, the modification does not add novel strongly unacceptable arguments, thus this procedure can be done iteratively and it is guaranteed that each strongly unacceptable argument is self-attacking after this modification. Next, we iteratively delete redundant and strongly defeated arguments. By Observation 7.14, the deletion of arguments does not introduce novel strongly unacceptable, redundant, or strongly defeated arguments. Moreover, by Lemma 7.17, redundant and strongly defeated arguments can be removed without producing novel redundant or strongly defeated arguments. □

The most sophisticated auxiliary result we require is that complete kernels of strongly equivalent cvAFs contain the same claims.

Lemma 7.19. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{co} \mathcal{G}$ implies $cl(A_{\mathcal{F}^{ck}}) = cl(A_{\mathcal{G}^{ck}})$.*

Proof. Consider an argument $x \in A_{\mathcal{F}^{ck}}$ with claim $cl(x) = c$. Towards a contradiction, assume that there is no argument $y \in A_{\mathcal{G}^{ck}}$ with $cl(y) = c$. We may assume $co_{cl}(\mathcal{F}^{ck}) = co_{cl}(\mathcal{G}^{ck})$, hence we deduce that x does not occur in any complete extension of $\mathcal{F}^{ck}$. Hence it does not occur in any admissible extension. Consequently, x receives incoming attacks.

Case 1 Suppose x is no self-attacker. The overall idea is as follows: We construct a set of instantiated arguments X in order to deal with all arguments that attack x. We introduce isolated arguments attacking (most of) them; this is possible due to our definition of the kernel. Then $f_e(\mathcal{F}^{ck}, X)$ has an admissible extension containing the argument x with claim c, where in $\mathcal{G}^{ck}$ claim c does not occur at all, showing that the two cvAFs cannot be strongly equivalent. So consider the set

$$\mathcal{Z} = \{z \in A_{\mathcal{F}^{ck}} \mid (z, x) \in R_\mathcal{F}\}$$

of arguments attacking x. Since x is no self-attacker, we have $vul(z) \not\subseteq vul(x)$, i.e., $vul(z) \setminus vul(x) \neq \emptyset$ for each $z \in \mathcal{Z}$ (otherwise, $vul(z) \subseteq vul(x)$ and $(z, x) \in R$ implies that x is strongly unacceptable, hence x would be self-attacking in the kernel). We let

$$V_\mathcal{Z} = \{v_e = (\emptyset, e) \mid e \in vul(z) \setminus vul(x), z \in \mathcal{Z}, e \neq c\},$$

i.e., we defeat these attackers as long as this would not require introducing claim c. Having c as claim, x can now defend itself, i.e., $\{x\} \cup V_\mathcal{Z}$ is admissible in the obtained cvAF. See Figure 3 for an example of the construction.

Since c does not occur in $\mathcal{G}^{ck}$ this is a witness for the absence of strong equivalence.

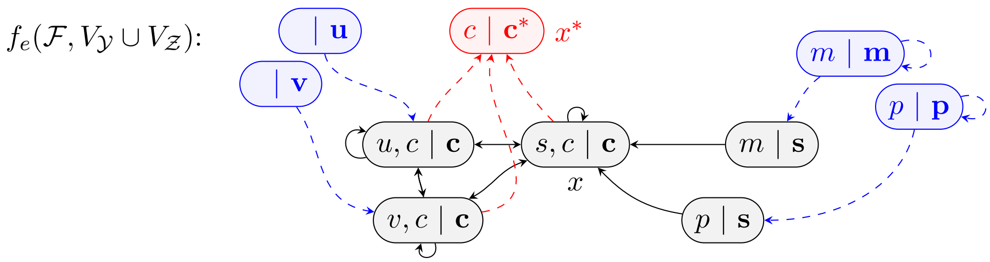


Figure 4: Illustration of Case 2.1 in the proof of Lemma 7.19. Novel arguments are in color with dashed attacks; left we depict arguments with claim c, i.e., the set $\mathcal{Y}$, and the novel arguments $V_\mathcal{Y}$ defeating them; right, we depict arguments attacking x and the novel self-attacking arguments which attack them. The novel argument x^* (in red) is undecided in the cvAF $\mathcal{F}$ and unattacked (hence accepted) in the cvAF $\mathcal{G}$.

Case 2 Now suppose each argument with claim c is a self-attacker and fix such x. Since x occurs in the kernel $\mathcal{F}^{ck}$, each attacker of x must itself possess attacking arguments.

The first step is to get rid of arguments with the same claim c. Consider the set

$$\mathcal{Y} = \{y \in A_{\mathcal{F}^{ck}} \mid cl(y) = c, y \neq x\}$$

of arguments with claim c. We consider arguments which defeat them; i.e., we let

$$\begin{aligned} V_\mathcal{Y} &= \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}, e \neq c\} \\ &= \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}\}. \end{aligned}$$

Now consider the set

$$\mathcal{Z} = \{z \in A_{\mathcal{F}^{ck}} \mid (z, x) \in R_{\mathcal{F}^{ck}}\} \setminus \mathcal{Y}$$

of arguments attacking x. We introduce self-attacking arguments that attack (most of) the arguments $z \in \mathcal{Z}$; i.e., we let

$$V_\mathcal{Z} = \{v_e = (\{e\}, e) \mid e \in vul(z), z \in \mathcal{Z}, e \neq c\}.$$

This ensures that all $z \in \mathcal{Z}$ with $vul(z) \neq \{c\}$ are undecided in the resulting cvAF.

Case 2.1: Suppose there is no argument z attacking x with $x \neq z$ and $vul(z) = \{c\}$, i.e., if $(z, x) \in R_{\mathcal{F}^{ck}}$, then $vul(z) \setminus \{c\} \neq \emptyset$. Hence introducing a self-attacker for each claim except c as done before ensures that x is undecided in each admissible extension; moreover, bear in mind that there is no other realization of c left after introducing $V_\mathcal{Y}$.

Now, consider some fresh argument $x^*_{c^*} = (\{c\}, c^*)$ with novel claim c^* which is attacked by c. This way, we ensure that $x^*_{c^*}$ is attacked by the (always undecided) self-attacker x in $f_e(\mathcal{F}^{ck}, X)$, but unattacked in $f_e(\mathcal{G}^{ck}, X)$. See Figure 4 for an illustrative example.

Case 2.2: Suppose there is some $z \in \mathcal{Z}$ attacking x with $x \neq z$ and $vul(z) = \{c\}$.

Suppose $cl(z) = d$ and consider

$$\mathcal{Y}_z = \{y \in A_{\mathcal{F}^{ck}} \mid cl(y) = d = cl(z)\}$$

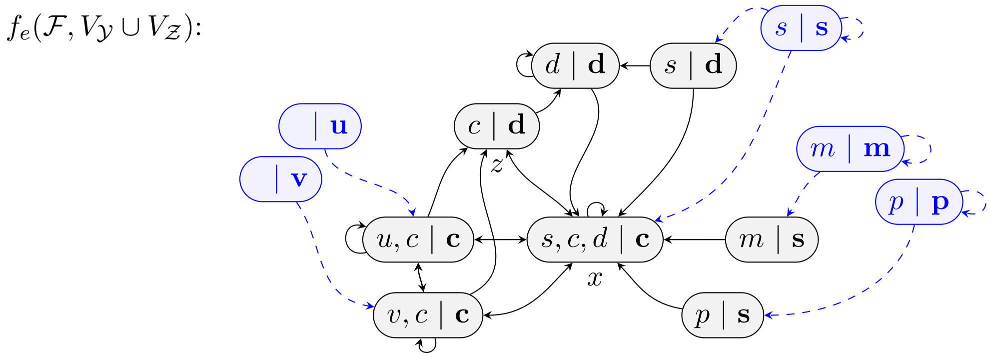


Figure 5: Illustration of Case 2.2 in the proof of Lemma 7.19. Novel arguments are in blue with dashed attacks. The argument $(\{c\}, d)$ is not grounded in the expansion of $\mathcal{F}$ but unattacked (thus grounded) in the expansion of $\mathcal{G}$.

Observe that $\mathcal{Y}_z \subseteq \mathcal{Z}$ (since $d \in vul(x)$ by assumption z attacks x). Hence for each $y \in \mathcal{Y}_z$, for each vulnerability $e \in vul(y)$ with $e \neq c$, we have introduced self-attacking arguments $(\{e\}, e)$ which attack y on e. Hence $z = (\{c\}, d)$ is the only argument with claim d which is not undecided (i.e., attacked by self-attacking arguments) in $f_e(\mathcal{F}^{ck}, V_\mathcal{Y} \cup V_\mathcal{Z})$. Hence there is no argument with claim d which is contained in the grounded extension of $f_e(\mathcal{F}^{ck}, V_\mathcal{Y} \cup V_\mathcal{Z})$. For an example of a cvAF $\mathcal{F}$ expanded by $V_\mathcal{Y} \cup V_\mathcal{Z}$ see Figure 5.

In $f_e(\mathcal{G}^{ck}, V_\mathcal{Y} \cup V_\mathcal{Z})$, on the other hand, the argument z is unattacked and thus contained in the grounded extension. □

We are now ready to prove the main result of this subsection: Indeed, the *co* kernel serves to characterize strong equivalence for two cvAFs w.r.t. *co* semantics.

Theorem 7.20. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{co} \mathcal{G}$ iff $\mathcal{F}^{ck} = \mathcal{G}^{ck}$.*

Proof. First assume $\mathcal{F}^{ck} = \mathcal{G}^{ck}$ holds. By Proposition 7.12, it holds that $\mathcal{F}^{ck} \equiv_s^{co} \mathcal{F}$ and $\mathcal{G}^{ck} \equiv_s^{co} \mathcal{G}$. Thus we obtain $\mathcal{F} \equiv_s^{co} \mathcal{G}$ by transitivity.

For the other direction, assume $\mathcal{F} \equiv_s^{co} \mathcal{G}$. We show that in this case, the kernels of $\mathcal{F}$ and $\mathcal{G}$ coincide. It suffices to show that they contain the same arguments, that is, we show that for all $x \in A_{\mathcal{F}^{ck}}$ there is $y \in A_{\mathcal{G}^{ck}}$ with $cl(y) = cl(x)$ and $vul(y) = vul(x)$.

By Lemma 7.19, $\mathcal{F}^{ck}$ and $\mathcal{G}^{ck}$ contain the same claims. We show that for all arguments x in $\mathcal{F}^{ck}$ there is some argument y in $\mathcal{G}^{ck}$ such that $cl(x) = cl(y) = c$ and $vul(y) \subseteq vul(x)$.

Let $x \in A_{\mathcal{F}^{ck}}$ with $cl(x) = c$. Then there is some argument y with claim c in $\mathcal{G}^{ck}$. Towards a contradiction, assume that for all $y \in A_{\mathcal{G}^{ck}}$ with $cl(y) = c$ we have $vul(y) \not\subseteq vul(x)$. Let $\mathcal{Y} = \{y \in A_{\mathcal{G}^{ck}} \mid cl(y) = c\}$ denote all arguments with claim c in $A_{\mathcal{G}^{ck}}$. Then for all $y \in \mathcal{Y}$ there is a claim $e \in vul(y)$ with $e \notin vul(x)$. We introduce arguments

$$V_\mathcal{Y} = \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}, e \neq c\}$$

in order to defeat all arguments in $\mathcal{G}^{ck}$ with claim c without introducing a novel argument with claim c. Now let $\mathcal{F}' = f_e(\mathcal{F}^{ck}, V_\mathcal{Y})$ and $\mathcal{G}' = f_e(\mathcal{G}^{ck}, V_\mathcal{Y})$.

Case 1 Suppose $c \in vul(x)$, i.e., x is self-attacking. Then each argument with claim c in $\mathcal{G}^{ck}$ is attacked by arguments in $V_\mathcal{Y}$. The cvAF $\mathcal{G}'$ has no argument with claim c since all such arguments are strongly defeated by $V_{\mathcal{Y}}$. On the other hand, x is contained in the kernel of $\mathcal{F}'$. By Lemma 7.19, $\mathcal{F}'$ and $\mathcal{G}'$ are not strongly equivalent to each other, contradicting our assumption.

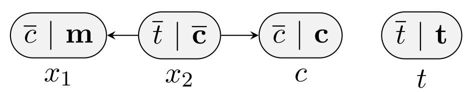
Case 2 Now assume x is not self-attacking. In this case, $\mathcal{G}'$ might still contain a single argument y with claim c and $vul(y) = vul(x) \cup \{c\}$. Thus the conclusion c does not appear in any conflict-free extension of $(\mathcal{G}')^{ck}$. We proceed analogously as in the proof of Lemma 7.19, Case 1, and introduce arguments to defend x in $(\mathcal{F}')^{ck}$ in order to guarantee that x appears in an admissible extension in the resulting cvAF. Then $\mathcal{F}^{ck}$ and $\mathcal{G}^{ck}$ do not yield the same admissible extensions after expansion.

We obtain that for every argument $x \in A_{\mathcal{F}^{ck}}$ there is exactly one argument $y \in A_{\mathcal{G}^{ck}}$ such that $cl(x) = cl(y)$ and $vul(x) = vul(y)$: Consider an argument $y \in A_{\mathcal{G}^{ck}}$ such that $cl(x) = cl(y) = c$ and $vul(x) \supseteq vul(y)$. By symmetry, there is $z \in A_{\mathcal{F}^{ck}}$ with $cl(z) = c$ such that $vul(y) \supseteq vul(z)$. Thus $vul(x) \supseteq vul(y) \supseteq vul(z)$. Since $\mathcal{F}^{ck}$ is redundancy-free, we obtain $vul(x) = vul(y) = vul(z)$. We conclude $x = z$ (by well-formedness, x and z attack the same arguments and are thus equivalent).

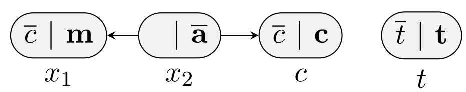
We thus obtain that $\mathcal{F}^{ck}$ and $\mathcal{G}^{ck}$ contain the same arguments in case $\mathcal{F}$ and $\mathcal{G}$ are strongly equivalent w.r.t. complete semantics. Since all attacks in cvAFs are determined by the claims and vulnerabilities of the arguments they contain, we conclude $\mathcal{F}^{ck} = \mathcal{G}^{ck}$. $\square$

Example 7.21. *Consider the two cvAFs*

$\mathcal{F}_D$:



$\mathcal{F}_{D'}$:



Recall that for X the set of all strongly unacceptable arguments in A we let

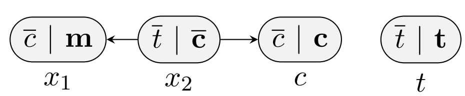
$$(A', R') = f_e(\mathcal{F} \setminus X, \{(vul(x) \cup \{cl(x)\}, cl(x)) \mid x \in X\}).$$

and then define

$$\begin{aligned} A^{ck} &= A' \setminus \{x \in A' \mid x \textit{ is strongly defeated or redundant}\}, \\ R^{ck} &= R' \cap (A^{ck} \times A^{ck}). \end{aligned}$$

The cvAF $\mathcal{F}_D$ does not contain any arguments of one of these kinds; hence $\mathcal{F}_D$ coincides with its own kernel, as we already discussed:

$\mathcal{F}_D^{ck}$:



In the complete kernel of $\mathcal{F}_{D'}$, on the other hand, two arguments are strongly defeated:

$\mathcal{F}_{D'}^{ck}$:

Clearly, $\mathcal{F}_D^{ck}$ and $\mathcal{F}_{D'}^{ck}$ differ and therefore, according to Theorem 7.20 the two cvAFs are not strongly equivalent w.r.t. complete semantics.

7.3 Preferred Kernel for cvAFs

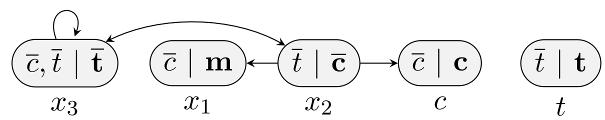
In this section, we present the preferred kernel $\mathcal{F}^{pk}$ for cvAFs which characterizes strong equivalence under preferred semantics. The main Theorem 7.26 of this section states that $\mathcal{F} \equiv_s^{pr} \mathcal{G}$ iff $\mathcal{F}^{pk} = \mathcal{G}^{pk}$. The proofs proceed similarly to the case for complete semantics and can be found in Appendix C. Below, we highlight the central concepts.

Towards a kernel for preferred semantics, we consider a special case of strong unacceptability that affects only preferred semantics.

Definition 7.22. *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 $vul(y) = \{cl(x)\}$. In this case, we call y the* witness *for strong pr-unacceptability of x.*

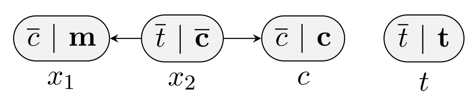
Since x is strongly unacceptable w.r.t. y we have $(y, x) \in R$ and $vul(y) \subseteq vul(x)$. From $\{cl(x)\} = vul(y) \subseteq vul(x)$ it follows that x is a self-attacker.

Example 7.23. *Consider the following cvAF $\mathcal{G}$*

$\mathcal{G}$:


Here x_3 is strongly pr-unacceptable: It is strongly unacceptable w.r.t. x_2 since x_2 attacks x_3 and has strictly fewer vulnerabilities since $vul(x_2) = \{\bar{t}\} \subseteq \{\bar{c}, \bar{t}\} = vul(x_3)$. Moreover, $vul(x_2) = \{\bar{t}\}$ which is indeed the claim of x_3.

Note the semantical intuition of this notion: Usually, a mutual attack between two arguments x and y leaves the AF with choices (take x, take y, or take none of them), but in the particular situation of pr-unacceptability, the pr-unacceptable argument does not contribute any choice. Indeed, $\mathcal{G}$ has the same preferred extensions as our running example $\mathcal{F}_D$

$\mathcal{F}_D$:


i.e., the absence of x_3 does not matter for preferred semantics.

Strongly pr-unacceptable arguments can always be removed without affecting the preferred extensions. Note that in contrast to usual strong unacceptability, the argument can be removed entirely, not just modified.

Proposition 7.24. *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 and can be computed by the following procedure: given a cvAF$\mathcal{F}$,

1. compute the complete kernel $\mathcal{F}^{ck}$
2. from the resulting cvAF remove all strongly pr-unacceptable arguments.

This procedure yields the preferred kernel $\mathcal{F}^{pk}$.

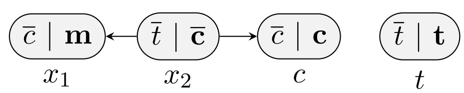
Definition 7.25. *For a cvAF* $\mathcal{F} = (A, R)$*, let* $\mathcal{F}^{ck} = (A^{ck}, R^{ck})$ *be as in Definition 7.10. We define the* preferred kernel $\mathcal{F}^{pk} = (A^{pk}, R^{pk})$ *with*

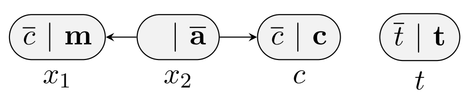
$$A^{pk} = A^{ck} \setminus \{x \in A^{ck} \mid x \textit{ is strongly } pr\textit{-unacceptable}\},$$
$$R^{pk} = R^{ck} \cap (A^{pk} \times A^{pk}).$$

To show that the preferred kernel characterizes strong equivalence for preferred semantics, we proceed as in the previous section. First we show that each cvAF is strongly equivalent to its preferred kernel. As a corollary, we obtain that the preferred extensions of a cvAF and its preferred kernel coincides. Moreover, the preferred kernel of a cvAF does neither contain redundant, non-self-attacking strongly unacceptable, strongly defeated or strongly pr-unacceptable arguments. We obtain our main result of this section:

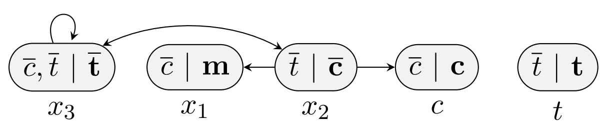
Theorem 7.26. *For two cvAFs* $\mathcal{F}$ *and* $\mathcal{G}$*,* $\mathcal{F} \equiv_s^{pr} \mathcal{G}$ *iff* $\mathcal{F}^{pk} = \mathcal{G}^{pk}$*.*

Example 7.27. *Consider the two cvAFs*

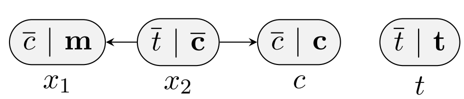
$\mathcal{F}_D$:


$\mathcal{F}_{D'}$:


and $\mathcal{G}$ *from above:*

$\mathcal{G}$:


For computing the preferred kernels, we first compute the complete kernels and the remove the pr*-unaccepted arguments. The cvAf* $\mathcal{F}_D$ *coincides with its complete kernel. Moreover, it has no strongly* pr*-unaccepted arguments and thus,* $\mathcal{F}_D^{pk} = \mathcal{F}_D$*.*

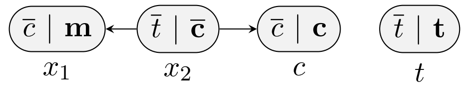
$\mathcal{F}_D^{pk}$:


After removing the strongly unaccepted arguments in $\mathcal{F}_{D'}$ *(complete kernel), we end up with* $\mathcal{F}_{D'}^{ck}$ *having no strongly* pr*-unaccepted argument, i.e.,* $\mathcal{F}_{D'}^{pk} = \mathcal{F}_{D'}^{ck}$*.*

$\mathcal{F}_{D'}^{pk}$:


The cvAF $\mathcal{G}$ *does not contain any strongly unacceptable arguments except* x_3 *which is already a self-attacker, i.e.,* $\mathcal{G}^{ck} = \mathcal{G}$*. As we already saw,* x_3 *is strongly* pr*-unacceptable in* $\mathcal{G}$ *and hence gets removed when computing the kernel:*

$\mathcal{G}^{pk}$:



Applying Theorem 7.26 yields strong equivalence w.r.t. preferred semantics of $\mathcal{F}_D$ and $\mathcal{G}$. None of those is strongly equivalent to the cvAF $\mathcal{F}_{D'}$.

7.4 Grounded Kernel for cvAFs

In this section, we present the grounded kernel $\mathcal{F}^{gk}$ for cvAFs which characterizes strong equivalence under grounded semantics. As in the previous subsection, the grounded kernel refines the complete kernel; in this case, by taking gr-unacceptable arguments into account. The main Theorem 7.30 of this section states that $\mathcal{F} \equiv_s^{gr} \mathcal{G}$ iff $\mathcal{F}^{gk} = \mathcal{G}^{gk}$. The proofs proceed similarly to the case for complete semantics and can be found in Appendix C.

First we recall our notion of strong gr-unacceptability of some argument x (see Definition 6.11). It states that there is $y \in A$ with $(y, x) \in R$ and $vul(y) \setminus \{cl(x)\} \subseteq vul(x)$. Analogously to strongly unacceptable arguments for the other semantics (see Proposition 7.9), we can turn these arguments into self-attackers.

Proposition 7.28. *Given a cvAF $\mathcal{F} = (A, R)$ and a strongly gr-unacceptable argument $x \in A$ and let $x' = (vul(x) \cup \{cl(x)\}, cl(x))$. Then $gr(\mathcal{F}) = gr((f_e(\mathcal{F} \setminus \{x\}, x'))$.*

The grounded kernel can be computed by the following procedure: given $\mathcal{F}$,

1. turn each strongly gr-unacceptable argument x into a self-attacker (i.e., formally add $cl(x)$ to the vulnerabilities $vul(x)$),

2. from the resulting cvAF remove all strongly defeated as well as redundant arguments.

Hence, it is defined analogously to the complete kernel by replacing X with the set of all strongly gr-unacceptable arguments in A.

Definition 7.29. *For a cvAF $\mathcal{F} = (A, R)$, let X denote the set of all strongly gr-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 grounded kernel $\mathcal{F}^{gk} = (A^{gk}, R^{gk})$ *with*

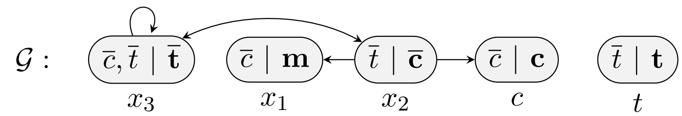
$$A^{gk} = A' \setminus \{x \in A' \mid x \textit{ is str. defeated or redundant}\},$$

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

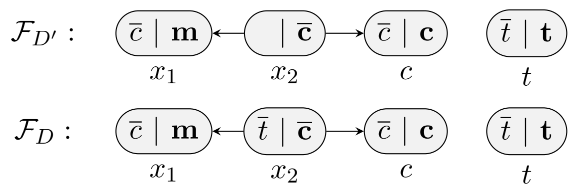
We proceed as for complete semantics. First we show that each cvAF is strongly equivalent to its grounded kernel. As a corollary, we obtain that the grounded extension of a cvAF and its grounded kernel coincides. Moreover, it holds that the grounded kernel of a cvAF does neither contain redundant nor strongly defeated arguments, and each strongly gr-unacceptable argument is self-attacking. With this, we can state the desired kernel characterization.

Theorem 7.30. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{gr} \mathcal{G}$ iff $\mathcal{F}^{gk} = \mathcal{G}^{gk}$.*

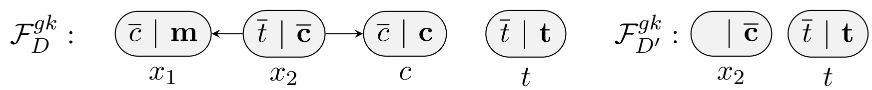
Example 7.31. *Consider our three cvAFs from before, i.e., the cvAF $\mathcal{G}$:*



and the two cvAFs $\mathcal{F}_D$ and $\mathcal{F}_{D'}$:



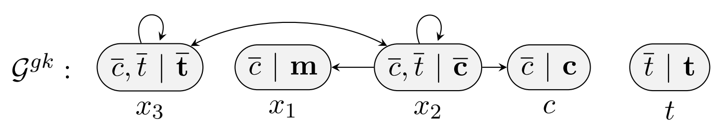
There is no strongly gr-unacceptable argument in $\mathcal{F}_D$ and $\mathcal{F}_{D'}$, so their gr-kernels coincide with the co-kernels:



In $\mathcal{G}$ we have that x_2 is strongly gr-unacceptable (although not strongly unacceptable): It holds that $(x_3, x_2) \in R_{\mathcal{G}}$ and

$$vul(x_3) \setminus \{cl(x_2)\} = \{\bar{c}, \bar{t}\} \setminus \{\bar{c}\} \subseteq \{\bar{t}\} = \{vul(x_2)\}.$$

Therefore, x_2 can be turned into a self-attacker and we obtain



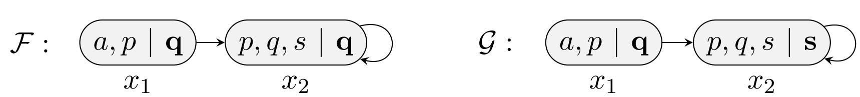
By Theorem 7.30, these cvAFs are pairwise not strongly equivalent w.r.t. grounded semantics.

7.5 Stable Kernel for cvAFs

In this section, we present the stable kernel $\mathcal{F}^{sk}$. The main Theorem 7.38 of this section states that $\mathcal{F} \equiv_s^{stb} \mathcal{G}$ iff $\mathcal{F}^{sk} = \mathcal{G}^{sk}$. The proofs proceed similarly to the case for complete semantics and can be found in Appendix C.

Strong equivalence w.r.t. stable semantics can be characterized in the same manner as in the previous subsections, but somewhat surprisingly, the corresponding kernel is by far the most involved one. We start with the crucial observation that the particular conclusion of self-attacking arguments is not of importance.

Example 7.32. *Consider the following two cvAFs $\mathcal{F}$ and $\mathcal{G}$:*



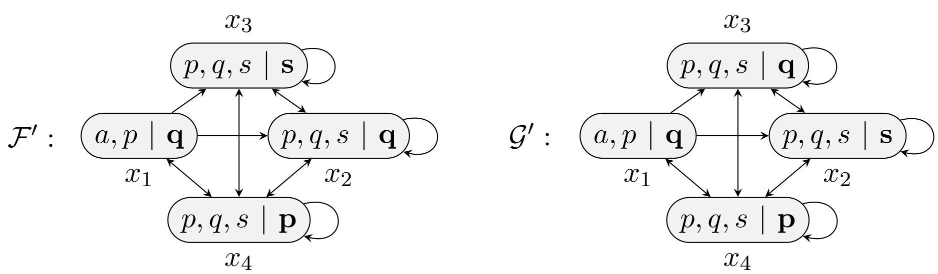
The only difference between $\mathcal{F}$ and $\mathcal{G}$ is the claim of the self-attacker x_2. Both $\mathcal{F}$ and $\mathcal{G}$ have the same unique stable extension $\{q\}$. As we will see, this is not a coincidence: for stable semantics, self-attacking arguments are indistinguishable w.r.t. their claims.

Proposition 7.33. *Given a cvAF $\mathcal{F} = (A, R)$ and a self-attacking argument $x \in A$. For any $s \in vul(x)$, it holds that $stb_{cl}(\mathcal{F}) = stb_{cl}(f_e(\mathcal{F}, \{(vul(x), s)\}))$.*

Proof. Let $\mathcal{F}' = f_e(\mathcal{F}, \{(vul(x), s)\})$ and let $y = (vul(x), s)$. Then $x, y \notin E$ for all stable extensions E in $\mathcal{F}$ and $\mathcal{F}'$. The statement thus follows by observing that y is attacked by a stable extension $E \in stb(\mathcal{F}')$ iff E attacks x in $\mathcal{F}'$ iff E attacks x in $\mathcal{F}$. □

Hence we can *add* all such self-attackers without changing stable semantics.

Example 7.34. *By adding all self-attackers $(vul(x_2), s)$ with $s \in vul(x_2)$ to our cvAFs $\mathcal{F}$ and $\mathcal{G}$ from Example 7.32 we obtain the following identical frameworks:*



Observe that for constructing the stable kernel, we have to *add* all of these self-attackers instead of deleting the initial one, even though this makes the resulting cvAF larger. The reason is that the self-attackers impose constraints on the stable extensions of the given framework, and thus need to be preserved. In order to ensure syntactical equivalence of the kernels, we have no choice but *adding* each version of a self-attacker, as demonstrated in the above example.

We also require a refinement of strong unacceptability called *strict strong unacceptability* to identify further arguments which can be deleted.

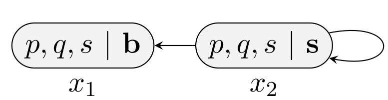
As shown in Proposition 7.7, strongly unacceptable arguments can be removed under stable semantics. However, as we observe in the above example, all of the arguments x_2, x_3, x_4 are strongly unacceptable w.r.t. to each other. Hence we consider the notion of *strictly* strongly unacceptable arguments to guarantee that our kernel is well-defined.

Definition 7.35. *For a cvAF $\mathcal{F} = (A, R)$, $x \in A$ is* strictly strongly unacceptable *if there is $y \in A$ with $(y, x) \in R$ and $vul(y) \subsetneq vul(x)$.*

To ensure that we catch all redundancies we need to take care of another issue which we illustrate in the following example.

Example 7.36. *Consider the cvAF $\mathcal{F}$ given as depicted below.*

$\mathcal{F}$:



The argument x_1 is not strictly strongly unacceptable w.r.t. x_2 hence it might be unsafe to remove it as observed above. However, if we apply the usual modification $x_1 \mapsto x_1'$ for strongly unacceptable arguments—adding the claim to the set of vulnerabilities—we obtain $vul(x_2) \subsetneq vul(x_1')$ for $x_1' = (\{p, q, s, b\}, b)$. Hence, the argument is now strictly strongly unacceptable w.r.t. x_2.

The many unacceptability and redundancy notions have to be treated carefully because they might interact with each other. Hence the order in which we proceed is crucial. To catch all redundancies we first have to add all 'missing' self-attackers. The stable kernel can be computed by the following procedure: given a cvAF $\mathcal{F}$,

1. turn each strongly unacceptable argument x into a self-attacker (i.e., formally add $cl(x)$ to the vulnerabilities $vul(x)$),
2. to the resulting cvAF, for each self-attacking argument x, add arguments $(vul(x), c)$ for all $c \in vul(x)$;
3. from the resulting cvAF remove all redundant, strongly defeated, and strictly strongly unacceptable arguments.

Formally, we construct the stable kernel as follows.

Definition 7.37. *For a cvAF $\mathcal{F} = (A, R)$, let X denote the set of all strongly unacceptable arguments in A and let*

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

Now, let Y denote the set of all self-attacking arguments in A and let

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

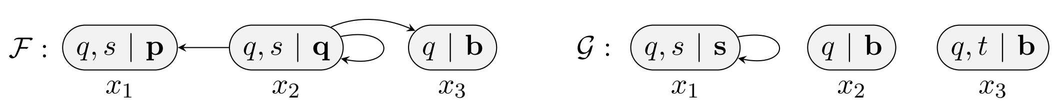
We define the stable kernel $\mathcal{F}^{sk} = (A^{sk}, R^{sk})$ *with*

$$A^{sk} = A' \setminus \{x \in A'' \mid x \text{ is str. defeated, strictly str. unacceptable, or redundant}\},$$
$$R^{sk} = R'' \cap (A^{sk} \times A^{sk}).$$

We are ready to present our characterization result for cvAF strong equivalence with respect to stable semantics (the proof proceeds analogously to the proof of Theorem 7.20).

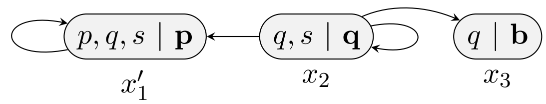
Theorem 7.38. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{stb} \mathcal{G}$ iff $\mathcal{F}^{sk} = \mathcal{G}^{sk}$.*

Example 7.39. *Consider the cvAFs $\mathcal{F}$ and $\mathcal{G}$ given as depicted below.*



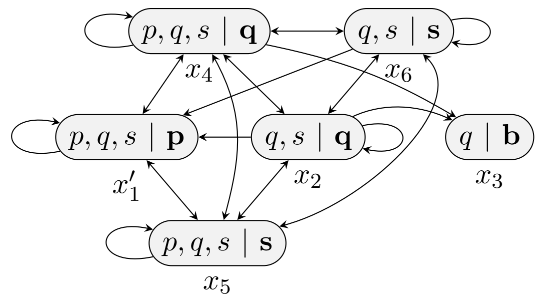
We construct the kernel of $\mathcal{F}$. First, we identify the set of all strongly unacceptable arguments $X = \{x_1\}$ of $\mathcal{F}$. We perform the first step in the kernel construction and turn x_1 into a self-attacker:

$\mathcal{F}'$:



Next, we add all missing self-attackers for each self-attacking argument in $Y = \{x'_1, x_2\}$*.*

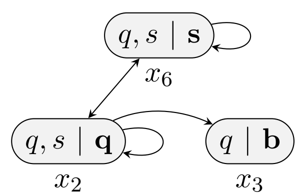
$\mathcal{F}''$:



Now, we identify all defeated, strictly strongly unacceptable, and redundant arguments and remove them. The argument x_4 *is redundant w.r.t.* x_2*: it has the same claim as* x_2 *and* $vul(x_2) \subset vul(x_4)$*. Likewise, the argument* x_5 *is redundant w.r.t.* x_6*. Moreover, the argument* x'_1 *is strictly strongly defeated by* x_2*: it is attacked by* x_2 *and* $vul(x_2) \subsetneq vul(x'_1)$*. Note that* x'_1 *is also strictly strongly defeated by* x_6 *(in fact, the arguments* x'_1*,* x_4*, and* x_5 *are strictly strongly defeated by both* x_2 *and* x_6*).*

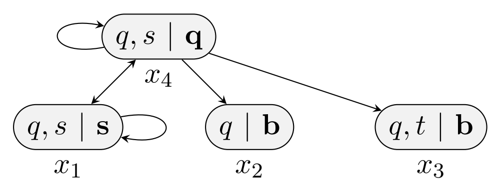
We remove the arguments x'_1*,* x_4*, and* x_5 *and obtain the following stable kernel of* $\mathcal{F}$*:*

$\mathcal{F}^{sk}$:



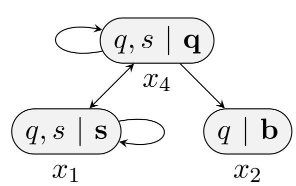
Now, let us construct the kernel of $\mathcal{G}$*. The cvAF has no strongly unacceptable arguments, hence we can directly proceed with the next step and add all missing self-attackers:*

$\mathcal{G}''$:



We proceed by removing all strongly defeated, strictly strongly unacceptable, and redundant arguments. In $\mathcal{G}$*, only the argument* x_3 *is redundant. Hence we obtain the following kernel:*

$\mathcal{G}^{sk}$:



Hence it turns out that this kernel coincides with the kernel of $\mathcal{F}$*, i.e.,* $\mathcal{F}^{sk} = \mathcal{G}^{sk}$*. By Theorem 7.38 conclude that* $\mathcal{F}$ *and* $\mathcal{G}$ *are strongly equivalent to each other w.r.t. stable semantics.*

7.6 Consequences for Assumption-based Argumentation

We have now established the desired (syntactical) strong equivalence characterizations for cvAFs. As it was the case for our enforcement notion, we first observe that deciding strong equivalence is intractable, even for atomic ABAs. More specifically, our construction used for Theorem 4.8 based on Reduction 4.4 yields an atomic ABA. So we obtain the following.

Corollary 7.40. *Deciding whether two given ABA frameworks D and D' are strongly equivalent* coNP*-hard even for atomic ABA frameworks.*

Therefore, we again focus on ABA frameworks with separated contraries. As we already observed in our discussion regarding enforcement, the way we tailored cvAFs ensures that these results are now ready to be applied to ABA. By transferring the above results in the context of ABA we obtain that deciding strong equivalence for atomic ABA frameworks with separated contraries is tractable.

For this, we first discuss a slight modification of the standard instantiation in which we instantiate proper arguments only, i.e, arguments whose claims are not assumptions.

Definition 7.41. *For an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$, we define the associated cvAF $\mathcal{F}_D^p = (A', R')$ by constructing (A, R) via Definition 5.7 and restrict the arguments to non-assumptions, i.e., $A' = A \setminus \{(X, p) \mid p \in \mathcal{A}\}$ and $R' = R \cap (A' \times A')$.*

We remark that even for a flat ABA framework D, the instantiations $\mathcal{F}_D$ and $\mathcal{F}_D^p$ differ: the cvAF $\mathcal{F}_D$ contains the arguments $(\{a\}, a)$ corresponding to assumptions $a \in \mathcal{A}$ since each assumption derives itself via $\{a\} \vdash_\emptyset a$. These arguments are removed in $\mathcal{F}_D^p$.

We call arguments that do not correspond to assumptions proper arguments. Likewise, we call conclusions of arguments that do not correspond to assumptions proper conclusions. Although the translation does not preserve the semantics in general, we obtain that for ABA frameworks with separated contraries, this is indeed the case.

Proposition 7.42. *For an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ with separated contraries, for $\sigma \in \{gr, co, pr, stb\}$, it holds that $\sigma_{cl}(\mathcal{F}_D^p) = \{S \setminus \mathcal{A} \mid S \in \sigma_{cl}(\mathcal{F}_D)\}$.*

Proof. Let a cvAF $\mathcal{F} = A, R)$ be arbitrary. We first show that (i) for any argument $a \in A$ with $a^+ = \emptyset$, any set of arguments $E \subseteq A \setminus \{a\}$, and semantics $\sigma \in \{cf, ad, gr, co, pr, stb\}$, it holds that $E \in \sigma(F \setminus \{a\})$ iff $E \in \sigma(F)$ or $E \cup \{a\} \in \sigma(F)$. We note that this holds true for each AF.

First, let $\sigma = cf$ and consider a set E not containing a. First, $E \cup \{a\} \in cf(F)$ implies $E \in cf(F)$; moreover, E is conflict-free in $F \setminus \{a\}$ iff E is conflict-free in F. If E is admissible, then it attacks the same arguments $b \in A \setminus \{a\}$ in F and $F \setminus \{a\}$. Moreover, since a has no outgoing attacks the statement extends to the set $E \cup \{a\}$, i.e., $E^+_{F \setminus \{a\}} = E^+_F \setminus \{a\} = (E \cup \{a\})^+_F \setminus \{a\}$. Since a has no outgoing attacks, it follows that E defends the same arguments $b \in A \setminus \{a\}$ in F and $F \setminus \{a\}$. Thus the statement holds true for admissible, complete, grounded, and preferred semantics. For stable semantics, we furthermore observe that removing a only causes the removal of a from the range of a stable set E; moreover, a is not undecided if a stable set exists in $F \setminus \{a\}$, thus the statement follows.

By observation (i), it holds that the removal of an argument $a \in A$ with no outgoing attacks in a given AF F corresponds to the removal of a from each extension E of F.

Coming back to our cvAF instantiation $\mathcal{F}_D$, we obtain $E \in \sigma(F_D \setminus A')$ iff $E \in \sigma(F_D)$ or $E \setminus A' \in \sigma(F_D)$ for $A' = \{a \in A \mid cl(a) \in \mathcal{A}\}$. Since we focus on flat ABA in this work, the set A' corresponds to the set $\{(\{a\}, a) \mid a \in \mathcal{A}\}$. Since D is flat by Assumption 2.3, the cvAF $\mathcal{F}_D \setminus A'$ does not contain any arguments with claims in $\mathcal{A}$. The result thus carries over to claim-level: $S \in \sigma_{cl}(\mathcal{F}_D \setminus A') = \sigma_{cl}(\mathcal{F}^p_D)$ iff $S \in \sigma_{cl}(\mathcal{F}_D)$ or $S \setminus \mathcal{A} \in \sigma_{cl}(\mathcal{F}_D)$ for each set of claims $S \subseteq cl(A)$. By Proposition 5.9, we have $\sigma_{Th}(D) = \sigma_{cl}(\mathcal{F}_D)$. The result follows when restricting $\sigma_{cl}(\mathcal{F}_D)$ to the set of proper conclusions. □

We are ready to prove the main theorem of this section, stating that deciding strong equivalence for two ABA frameworks within our usual fragment is tractable. Our goal is to apply our kernel characterizations for cvAFs. Our first step is to handle the assumption arguments $\{(\{\overline{a}\}, a) \mid a \in \mathcal{A}\}$ manually. Afterwards, we move from $\mathcal{F}_D$ to $\mathcal{F}^p_D$ and thereby apply Proposition 7.42. Then we are ready to utilize our cvAF results.

Theorem 7.43. *For two atomic ABA frameworks D and D' with separated contraries, deciding $D \equiv^\sigma_s D'$ is tractable.*

Proof. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ and $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}', ^{-\prime})$. We construct their corresponding cvAFs $\mathcal{F}_D$ and $\mathcal{F}_{D'}$.

Let us first consider the instantiated arguments corresponding to assumptions, i.e., $X^D = \{(\{\overline{a}\}, a) \mid a \in \mathcal{A}\}$ in $\mathcal{F}_D$ resp. $X^{D'} = \{(\{\overline{a}\}, a) \mid a \in \mathcal{A}'\}$ in $\mathcal{F}_{D'}$. We make the following observation: for each assumption-argument $x \in X^D \cup X^{D'}$, it holds that x is either i) strongly defeated or ii) strongly unacceptable, or iii) remains unchanged in the kernel of the corresponding cvAF. Let us discuss all other cases.

- In case x is strictly strongly unacceptable, it is strongly defeated: because $vul(x) = \{\overline{a}\}$ is a singleton, strict unacceptability implies that x is attacked by some argument with no vulnerabilities.
- Strong gr-unacceptability is equivalent to strong unacceptability for ABA frameworks which separate contraries; hence it suffices to discuss the latter.
- It cannot be redundant because $cl(x)$ does appear as conclusion of some other argument (we consider flat ABA frameworks).
- It cannot be strongly pr-unacceptable because $cl(x)$ cannot appear as vulnerability of any argument since we assume that D and D' separate contraries.

So let us consider the cases i), ii), and iii) mentioned above.

i) Suppose $x \in X^D \cup X^{D'}$ is strongly defeated. By our previous results, we can remove the assumption from the corresponding ABA framework without changing the semantics (even considering arbitrary expansions). Hence, we can w.l.o.g. assume that no assumption is strongly defeated in D or D'.

ii) Let X^D_{su} and $X^{D'}_{su}$ denote the set of assumption-arguments that are strongly unacceptable in $\mathcal{F}_D$ and $\mathcal{F}_{D'}$, respectively. We show that $X^D_{su} \neq X^{D'}_{su}$ implies that $D \not\equiv^\sigma_s D'$ and hence, we can assume $X^D_{su} = X^{D'}_{su}$. By symmetry, it suffices to consider some $x \in X^D_{su} \setminus X^{D'}_{su}$

First note that if $x = (\{\overline{a}\}, a) \in X^D_{su}$ is strongly unacceptable, there must be some argument y occurring in $\mathcal{F}_D$ which attacks x (i.e., $cl(y) = \overline{a}$) and satisfies $vul(y) \subseteq vul(x)$ (i.e., $vul(y) = \emptyset$ or $vul(y) = \{\overline{a}\}$), where the case $vul(y) = \emptyset$ is excluded since x is not strongly defeated. Thus, y is of the form $y = (\{\overline{a}\}, \overline{a})$.

Since D is atomic, the only way to induce such an argument y is due to the rule $\overline{a} \leftarrow a$. If this rule occurs in D' as well, then $a \in \mathcal{A}'$ since D' is atomic; thus $x \in X^{D'}_{su}$ contradicting our assumption. So $\overline{a} \leftarrow a.$ does not occur in D', and we proceed as follows:

(a) Suppose $\overline{a} \in Th_{D'}(\emptyset)$ (i.e., $\overline{a}$ must be a fact since D' is atomic). Then $\overline{a}$ is a fact in D', but by assumption not in D. Then consider

$$\mathcal{R}_H = \{\overline{b} \leftarrow . \mid b \in (A \cup \mathcal{A}') \setminus \{a\}\}$$

and let $H = \{\mathcal{L} \cup \mathcal{L}', \mathcal{R}_H, \mathcal{A} \cup \mathcal{A}', {}^{-} \cup {}^{-\prime}\}$. This ABA framework shows that D and D' are not strongly equivalent ($\overline{a}$ is accepted in $D' \cup H$, but not in $D \cup H$; for any semantics considered in this paper).

(b) Suppose $\overline{a} \notin Th_{D'}(\emptyset)$; and assume for the moment $a \in \mathcal{A}'$. Since the rule $\overline{a} \leftarrow a.$ does not occur in D', our reasoning from above shows that a is neither strongly unacceptable nor strongly defeated in D' (and the other cases cannot occur). Therefore, our enforcement results show that a can be enforced in D', but not in D (yielding a suitable counter-example for strong equivalence). Finally, if $a \notin \mathcal{A}'$, then first add $H = (\{a\}, \emptyset, \{a\}, \{a \mapsto \overline{a}\})$ and apply the same argument afterwards.

Hence $X^D_{su} \neq X^{D'}_{su}$ implies that $D \not\equiv^\sigma_s D'$, i.e., we can assume $X^D_{su} = X^{D'}_{su}$.

iii) Let X^D_n and $X^{D'}_n$ denote the set of assumption-arguments that remain unchanged in the kernel of $\mathcal{F}_D$ and $\mathcal{F}_{D'}$, respectively. By our enforceability results we can enforce each of them, hence we immediately obtain that $X^D_n \neq X^{D'}_n$ implies that $D \not\equiv^\sigma_s D'$.

To summarize, we may w.l.o.g. assume $\mathcal{A} = \mathcal{A}'$, otherwise we can handle the ABA frameworks with the above arguments. Now we are ready to apply our cvAF results. Analogously as for AFs, we write $\mathcal{F}^{k(\sigma)}$ to denote the kernel which characterize strong equivalence for the semantics σ. Given $\mathcal{A} = \mathcal{A}'$, the following holds.

$$\begin{aligned}
(\mathcal{F}^p_D)^{k(\sigma)} = (\mathcal{F}^p_{D'})^{k(\sigma)} &\Leftrightarrow \mathcal{F}^p_D \equiv^\sigma_s \mathcal{F}^p_{D'} && (1)\\
&\Leftrightarrow \text{for each set of inst. args } X : \sigma_{cl}(f_e(\mathcal{F}^p_D, X)) = \sigma_{cl}(f_e(\mathcal{F}^p_{D'}, X)) && (2)\\
&\Leftrightarrow \text{for each ABA } H : \sigma_{cl}(\mathcal{F}^p_{D\cup H}) = \sigma_{cl}(\mathcal{F}^p_{D'\cup H}) && (3)\\
&\Leftrightarrow \text{for each ABA } H : \sigma_{cl}(\mathcal{F}_{D\cup H}) = \sigma_{cl}(\mathcal{F}_{D'\cup H}) && (4)\\
&\Leftrightarrow \text{for each ABA } H : \sigma_{Th}(D \cup H) = \sigma_{Th}(D' \cup H) && (5)\\
&\Leftrightarrow D \equiv^\sigma_s D' && (6)
\end{aligned}$$

(2) By Theorems 7.20, 7.30, 7.26, and 7.38 for the respective semantics.

(3) By definition of strong equivalence for cvAFs.

(4) The crucial observation is that each rule r with assumptions that appear in the frameworks at hand corresponds to an instantiated argument and vice versa.

($\Rightarrow$) Given ABA $H = (\mathcal{L}, \mathcal{R}'', \mathcal{A}'', ^{-})$, we let $X = \{(\overline{A}, p) \mid p \leftarrow A \in \mathcal{R}'', A \subseteq \mathcal{A} \cup \mathcal{A}''\}$. By Lemma 5.15, it holds that $\mathcal{F}^p_{D\cup\{r\}} = f_e(\mathcal{F}^p_D, \{(\overline{A}, p)\})$ for each rule $r = p \leftarrow A$ with $A \subseteq \mathcal{A} \cup \mathcal{A}''$. We obtain $\mathcal{F}^p_{D\cup H} = f_e(\mathcal{F}^p_D, X)$ and $\mathcal{F}^p_{D'\cup H} = f_e(\mathcal{F}^p_{D'}, X)$. Since $\sigma_{cl}(f_e(\mathcal{F}^p_D, X)) = \sigma_{cl}(f_e(\mathcal{F}^p_{D'}, X))$ we thus obtain $\sigma_{cl}(\mathcal{F}^p_{D\cup H}) = \sigma_{cl}(\mathcal{F}^p_{D'\cup H})$.

($\Leftarrow$) Given a set of arguments X, we consider an expansion $H = (\mathcal{L}, \mathcal{R}'', \mathcal{A}'', ^{-})$ such that all arguments in X are instantiated. For this, we need to ensure that $D \cup H$ contains all necessary assumptions, that is, we let $\mathcal{A}'' = \bigcup_{(\overline{A},p)\in X} A$. Now, we add a rule for each argument in X, i.e., we let $\mathcal{R}'' = \{p \leftarrow A \mid (\overline{A}, p) \in X\}$. By Lemma 5.15, we obtain $\mathcal{F}^p_{D\cup H} = f_e(\mathcal{F}^p_D, X)$. Thus the statement follows.

(5) ($\Rightarrow$) Consider a ABA H. Let $H' = H \cup H_{\mathcal{A}}$ where $H_{\mathcal{A}}$ is the ABA framework consisting of the assumptions (and their contraries), i.e., $H_{\mathcal{A}} = (\mathcal{A} \cup \overline{\mathcal{A}}, \emptyset, \mathcal{A}, ^{-})$. By our assumption it holds that $\sigma_{cl}(\mathcal{F}^p_{D\cup H'}) = \sigma_{cl}(\mathcal{F}^p_{D'\cup H'})$. Hence we can add the assumptions to the instantiation: it holds that

$$\sigma_{cl}(\mathcal{F}^p_{D\cup H\cup H_{\mathcal{A}}}) = \sigma_{cl}(\mathcal{F}_{D\cup H}) = \sigma_{cl}(\mathcal{F}_{D'\cup H}) = \sigma_{cl}(\mathcal{F}^p_{D'\cup H\cup H_{\mathcal{A}}}).$$

($\Leftarrow$) By Proposition 7.42, we can remove the assumptions from the extensions.

(6) By definition of strong equivalence for ABA frameworks.

Thus, to decide strong equivalence between D and D', it suffices to check

(i) $\mathcal{A} \setminus \{a \in \mathcal{A} \mid \overline{a} \leftarrow \in \mathcal{R}\} = \mathcal{A}' \setminus \{a \in \mathcal{A}' \mid \overline{a} \leftarrow \in \mathcal{R}'\}$; if this is not the case, we have $D \not\equiv^{\sigma}_{s} D'$; otherwise, we check

(ii) syntactical equivalence of the σ-kernels of $\mathcal{F}^p_D$ and $\mathcal{F}^p_{D'}$. □

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

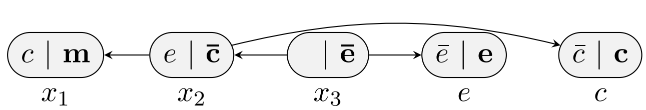
As a final remark in this section, let us head back to Jane and Antoine.

Example 7.44. *Recall that the ABA framework where Antoine used the excuse of having no money was given via*

$$\mathcal{L} = \{c,\ \overline{c},\ e,\ \overline{e},\ m\} \qquad \mathcal{A} = \{c,\ e\} \qquad \mathcal{R} = \{m \leftarrow c.,\quad \overline{c} \leftarrow e.,\quad \overline{e} \leftarrow .\}$$

and induced the instantiated cvAF

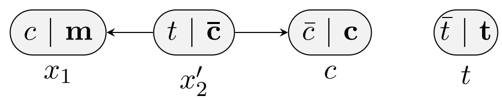
$\mathcal{F}_D$:



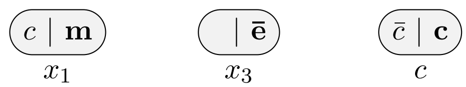
On the other hand, the "trailer" version was given as

$$\mathcal{L}' = \{c,\ \bar{c},\ m,\ t,\ \bar{t}\} \qquad \mathcal{A}' = \{c,\ t\} \qquad \mathcal{R}' = \{m \leftarrow c.,\quad \bar{c} \leftarrow t.\}.$$

with cvAF

$\mathcal{F}_{D'}$:


Since Jaine's voucher argument x_3 renders x_2 and e strongly unacceptable, for all considered semantics the kernel is given as

$\mathcal{F}_D^{k(\sigma)}$:


In the second version, there is no argument to be modified or removed, i.e., $\mathcal{F}_{D'}^{k(\sigma)} = \mathcal{F}_{D'}$. Hence, w.r.t. no semantics, these two discussions correspond to strongly equivalent ABA frameworks, which matches our intuition.

8. Tractability Results for Logic Programs

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 at 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.

8.1 Background

A logic program P consist of rules of the form

$$r:\ c \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m \tag{7}$$

where $0 \leq n, m$ and the a_i, b_i and c are ordinary atoms. We let $head(r) = c$, $pos(r) = \{a_1, \ldots, a_n\}$ and $neg(r) = \{b_1, \ldots, b_m\}$; $\mathcal{L}(P)$ is the set of all atoms occurring in P. Given a rule r and an atom a, we write $r \cup \{\texttt{not}\ a\}$ to denote the rule that results from the addition of a to $neg(r)$. For a set $B = \{b_1, \ldots, b_n\}$ of atoms we write $\texttt{not}\ B$ as a shorthand for the conjunction $\texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$. If $pos(r) = neg(r) = \emptyset$ we simply write $head(r)$. to denote the rule and call it a *fact*.

Let us now define the semantics of an LP P. The first step is the notion of an interpretation $I = (T, F)$ where one can intuitively think of "true" and "false" atoms, respectively.

Definition 8.1. *A 3-valued Herbrand Interpretation I of an LP P is a tuple $I = (T, F)$ with $T \cup F \subseteq \mathcal{L}(P)$ and $T \cap F = \emptyset$.*

The *reduct* P/I if an LP P w.r.t. some interpretation I evaluates the default negated literals according to I, where the body of each rule is interpreted as a conjunction: i) If a rule contains "$\texttt{not}\ b$" for some $b \in T$, then the rule is not applicable; ii) each occurrence of "$\texttt{not}\, b$" for some $b \in F$ is evaluated to true and can be removed from the corresponding body; iii) otherwise, if $b \notin T \cup F$, then it is undecided - and so is "$\texttt{not}\, b$".

Formally, given P with interpretation $I = (T, F)$ we define the *reduct* P/I of P w.r.t. I as follows: Starting from P,

i) remove each rule r from P with $T \cap neg(r) \neq \emptyset$,

ii) remove "$\texttt{not}\, b$" from each remaining rule whenever $b \in F$, and

iii) replace each occurrence of "$\texttt{not}\, b$" from each remaining rule with a fresh atom u.

Example 8.2. *Let P be the LP given as follows.*

$$P: \; a \leftarrow \textit{not}\, b. \qquad b \leftarrow \textit{not}\, a. \qquad c \leftarrow b. \qquad d \leftarrow a, \textit{not}\, e. \qquad e \leftarrow a, \textit{not}\, d.$$

For $I = (T, F)$ with $T = \{a\}$ and $F = \{b, c\}$ the reduct P/I is given as

$$P/I: \; a \leftarrow . \qquad c \leftarrow b. \qquad d \leftarrow a, u. \qquad e \leftarrow a, u.$$

By $\Psi_P(I) = (T_\Psi, F_\Psi)$ we denote the least 3-valued model of P/I, i.e., T_Ψ is minimal and F_Ψ maximal s.t.

a) $a \in T_\Psi$ iff there is a rule $r \in P/I$ with $a \in head(r)$ and $pos(r) \subseteq T_\Psi$,

b) $a \in F_\Psi$ iff for each rule $r \in P/I$ with $a \in head(r)$ we have $pos(r) \cap F_\Psi \neq \emptyset$.

Example 8.3. *For the previously computed reduct*

$$P/I: \; a \leftarrow . \qquad c \leftarrow b. \qquad d \leftarrow a, u. \qquad e \leftarrow a, u.$$

we have $\Psi_P(I) = (T_\Psi, F_\Psi) = (\{a\}, \{b, c\})$.

Definition 8.4. *A 3-valued interpretation $I = (T, F)$ of P is*

- *P-stable, $I \in pstb(P)$, if $I = \Psi_P(I)$;*
- *well-founded, $I \in wf(P)$, if I is P-stable with minimal T;*
- *regular, $I \in reg(P)$, if I is P-stable with maximal T;*
- *stable, $I \in stb(P)$, if I is P-stable and $T \cup F = \mathcal{L}(P)$.*

Since stable models are two-valued, we sometimes identify I with T, i.e., we write $T \in stb(P)$ whenever there is a stable model $I = (T, F)$ of P.

Example 8.5. *For our program from above $I = (\{a\}, \{b, c\})$ is a P-stable model; it is not stable since c and d are neither in T nor F (interpreted as "undefined"). The reader may check that $I' = (\{a, d\}, \{b, c, e\})$ is a stable model of P.*

Atomic LPs (Janhunen, 2004) are similar in spirit to atomic ABA frameworks.

Definition 8.6. *A rule r of the form* (7) *is called* atomic *if $pos(r) = \emptyset$; a logic program P is* atomic *if each $r \in P$ is.*

8.2 LPs and cvAFs

LPs can be translated into AFs and vice versa. Let us briefly recall the most relevant results from the literature. Given an LP P, we can construct a corresponding AF as follows (Caminada et al., 2015b).

Definition 8.7. *For an LP P, A is an argument in P, denoted by $A \in Args(P)$, with*

- $\textsc{Conc}(A) = c$,
- $\textsc{Rules}(A) = \bigcup_{i \leq n} \textsc{Rules}(A_i) \cup \{r\}$, *and*
- $\textsc{Vul}(x) = \bigcup_{i \leq n} \textsc{Vul}(A_i) \cup \{b_1, \dots, b_m\}$

if and only if there are $A_1, \dots, A_n \in Args(P)$ and a rule $r \in P$ with

$$r = c \leftarrow \textsc{Conc}(A_1), \dots, \textsc{Conc}(A_n), \mathit{not}\ b_1, \dots, \mathit{not}\ b_m.$$

and $r \notin \textsc{Rules}(A_i)$ for all $i \leq n$. An argument A attacks another argument B if $\textsc{Conc}(A) \in \textsc{Vul}(B)$. All arguments and the induced attacks form the corresponding AF $F_P = (A_P, R_P)$.

Naturally, we obtain a cvAF $\mathcal{F}_P$ corresponding to some LP P by constructing an instantiated argument $(\textsc{Vul}(A), \textsc{Conc}(A))$ for each $A \in Args(P)$ as well as the induced attack relation. It has been shown that the aforementioned translation to AFs preserves the semantics of the underlying LP (Caminada et al., 2015b). For cvAFs, we get this result from corresponding observations for claim-augmented AFs (König, Rapberger, & Ulbricht, 2022, Proposition 4.5); although the mentioned result does not make any statement about cvAFs, the translation is the same except that we add the vulnerabilities explicitly. We thus obtain:

Proposition 8.8. *Let P be an LP and $\mathcal{F}_P$ the corresponding cvAF. If $I = (T, F)$ is*

- *stable in P, then $T \in stb(\mathcal{F}_P)$,*
- *P-stable in P, then $T \in co(\mathcal{F}_P)$,*
- *regular in P, then $T \in pr(\mathcal{F}_P)$,*
- *well-founded in P, then $T \in gr(\mathcal{F}_P)$,*

Vice versa, given a cvAF $\mathcal{F} = (A, R)$ we define the corresponding LP $P_\mathcal{F}$ as the set

$$P_\mathcal{F} = \{\textsc{Conc}(x) \leftarrow \texttt{not}\ \textsc{Vul}(x) \mid x \in A\}$$

of rules. Again we can borrow from previous research (König et al., 2022, Proposition 4.5).

Proposition 8.9. *Let $\mathcal{F} = (A, R)$ be a cvAF and $P_\mathcal{F}$ the corresponding LP. If $E \subseteq A$ is*

- *stable in $\mathcal{F}$, then there is some $I \in stb(P_\mathcal{F})$ of the form $I = (E, F)$,*
- *complete in $\mathcal{F}$, then there is some $I \in pstb(P_\mathcal{F})$ of the form $I = (E, F)$,*
- *preferred in $\mathcal{F}$, then there is some $I \in reg(P_\mathcal{F})$ of the form $I = (E, F)$,*
- *grounded in $\mathcal{F}$, then there is some $I \in wf(P_\mathcal{F})$ of the form $I = (E, F)$.*

8.3 Dynamics in LPs

Let us now turn to our dynamic scenarios. We proceed as for ABA frameworks by applying the cvAF results. We give an LP version of Lemma 5.15.

Lemma 8.10. *Given an atomic LP P.*

- *For each atomic rule $r = c \leftarrow \texttt{not}\ B$, we have $\mathcal{F}_{P \cup \{r\}} = f_e(\mathcal{F}_P, x)$ 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 $r = c \leftarrow \texttt{not}\ B$.*

We are now ready to efficiently investigate our two problems we considered before. The relation is even much closer since we do not need to handle additional assumptions. In accordance with our general definitions of enforcement and strong equivalence in non-monotonic reasoning formalisms, we define the LP enforcement problem as follows.

Definition 8.11. *Let P be an LP and σ a semantics. An atom p is σ-*enforceable *if there is a set R of rules s.t. $head(r) \neq p$ for all $r \in R$ and p is credulously accepted in $P \cup R$ w.r.t. semantics σ.*

As for ABA, we get intractability in the general case.

Proposition 8.12. *Consider a semantics σ. Deciding atom-enforceability w.r.t. σ for the class of normal LPs is* NP*-hard.*

Proof. Let φ be a boolean formula given by clauses C over variables in X. The corresponding logic program P contains the following rules:

- the atomic rule '$p_\varphi \leftarrow \texttt{not}\ C$';
- rules '$p_c \leftarrow \{l \mid \neg l \in c\}, \texttt{not}\ \{l \in X \mid l \in c\}$' for each clause $c \in C$.

Intuitively, a clause-atom c is contained in a stable model M iff c is false in M. Hence we can accept φ iff $c \notin M$ for all $c \in C$. It can be shown that φ is satisfiable iff p_φ is enforceable in P (proof details can be found in Appendix D). □

Our cvAF results yield tractability for atomic LPs. We can apply our results directly.

Theorem 8.13. *For atomic LPs, deciding whether some atom is enforceable is tractable.*

Proof. By Corollary 6.13, we have for any atom a: a is enforceable in P iff a is credulously accepted in $P \cup H$ for some H iff a is credulously accepted in $f_e(\mathcal{F}_P, X)$ for some X iff a is enforceable in $\mathcal{F}_P$. By Theorem 6.9 and Proposition 6.12, the latter is tractable. □

Let us next discuss strong equivalence for atomic LPs. In general, we define strong equivalence for LP relative to a LP-fragment $\mathfrak{C}$ as follows.

Definition 8.14. *Two LPs $P, P' \in \mathfrak{C}$ are* strongly equivalent *w.r.t. a semantics σ in the fragment $\mathfrak{C}$, for short $P \equiv^\sigma P'$, if for each LP $R \in \mathfrak{C}$, it holds that $\sigma(P \cup R) = \sigma(P' \cup R)$.*

Without the requirement of P, P', and R being atomic, intractability of strong equivalence is well-known (Pearce et al., 2001; Lin, 2002). Due to our cvAF results, we obtain a tractable fragment here as well.

Theorem 8.15. *Deciding strong equivalence in the class of atomic LPs is tractable.*

Proof. Immediate from Theorems 7.20, 7.30, 7.26, and 7.38: for two LPs P and P' it holds that P is atomic strongly equivalent to P' iff $\sigma(P \cup R) = \sigma(P' \cup R)$ for each atomic set of rules R iff $\sigma(f_e(\mathcal{F}_P, H)) = \sigma(f_e(\mathcal{F}_{P'}, H))$ for $H = \bigcup\{(B, c) \mid c \leftarrow \texttt{not}\ B \in R\}$ for each R iff $\sigma(f_e(\mathcal{F}_P, X)) = \sigma(f_e(\mathcal{F}_{P'}, X))$ for each set X of instantiated arguments iff $\mathcal{F}_P^{k(\sigma)} = cvF_{P'}^{k(\sigma)}$, where $\mathcal{F}^{k(\sigma)}$ denotes the kernel which characterize strong equivalence for semantics σ. □

For stable model semantics, we obtain an even more general result: strong equivalence between two atomic LPs is tractable even if we consider expansions with rules that are non-atomic. For this, we will first show that each atomic LP P is strongly equivalent to the program obtained by re-translating the stable kernel $\mathcal{F}_P^{sk}$ w.r.t. stable semantics.

Proposition 8.16. *Let P be an atomic LP and let P^{sk} denote the logic program $P_{\mathcal{F}_P^{sk}}$. It holds that P and P^{sk} are strongly equivalent w.r.t. stable semantics.*

Proof. In the following, we use the terms instantiated arguments and atomic rules interchangeably. For simplicity, we will talk about redundant, strongly unacceptable, and strongly defeated rules instead of formally switching between the formalisms. By Lemma 8.10, these concepts are indeed transferrable to the realm of atomic LPs.

Consider a set H of rules. We show that M is a stable model of $P' = P \cup H$ iff M is a stable model of $P'' = P^{sk} \cup H$. The underlying observation is that the reduct of P'/M coincides with P''/M in case M is a model of P' or P''.

($\Rightarrow$) First assume M is a stable model of P', i.e., let $M \in stb(P \cup H) = stb(P')$.

We show that $P'/M = P''/M$. We first observe that each rule $r \in H$ undergoes the same changes in P'/M and P''/M because the set of true facts (M) coincides in both reducts. That is, for each rule $r \in H$, we have $r' = r''$ where $r' \in P'/M$ denotes the modification of r in P'/M, and r'' denotes the modification of r in P''/M (in case $M \cap neg(r) = \emptyset$), and $r \notin P'/M$ iff $r \notin P''/M$ (if $M \cap neg(r) \neq \emptyset$). Moreover notice that all other rules not originating from rules in H are facts because P is atomic. Assume $a. \in P'/M$ but $a. \notin P''/M$. Let $r \in P$ denote some rule with $head(r) = a$ which has survived the reduct modifications. That is, each negated literal in the body of r is false, i.e.,

$$neg(r) \cap M = \emptyset. \tag{8}$$

Now since $a. \notin P''/M$ we have either (1) r is deleted when building the kernel of P or (2) r is strongly unacceptable in P and thus the modified rule $r' = a \leftarrow neg(r) \cup \{\texttt{not}\ a\}$ is deleted when building the reduct of $P^{sk} \cup H$.

- Let us first deal with <u>case 2:</u> let $t \in P$ be a rule witnessing unacceptability of r in P. That is, $neg(t) \subseteq neg(r)$ and $head(t) \in neg(r)$. We therefore infer from (8)

  $$neg(t) \cap M \subseteq neg(r) \cap M = \emptyset,$$

  hence the rule $head(t).$ is contained in P'/M. Since M is a stable model of P', it holds that $head(t) \in M$. Consequently, $M \cap neg(r) \neq \emptyset$ contradicting our above assumption (8), i.e., this case cannot occur.

- In case 1, rule r is deleted when constructing the kernel P^{sk}. That is, r is either i) strongly defeated, ii) strictly strongly unacceptable, or iii) redundant in P.

  i) In the former case, there is some fact $b. \in P$ such that $b \in neg(r)$. Hence $b \in M$ and we obtain that r is deleted when constructing the reduct P'.

  ii) In case r is strictly strongly unacceptable, we proceed as in case 2.

  iii) In case r is redundant, consider a rule $s \in P$ with $neg(s) \subsetneq neg(r)$ and $head(r) = head(s)$. W.l.o.g., let s be minimal in that aspect (i.e., there is no rule s' with $neg(s') \subsetneq neg(r)$ and $head(r) = head(s')$ and $neg(s') \subsetneq neg(s)$). If s is contained in P^{sk}, then we have $neg(s) \cap M = \emptyset$ (due to (8)) and $head(s) = a$, hence we have found a witness showing that the fact $a.$ is contained in P'' as well. In case s is not contained in P^{sk}, it holds that s is either strictly strongly unacceptable (we proceed as in case 2) or strongly defeated (we proceed as above).

  Hence we obtain that $P'/M \subseteq P''/M$.

For the other direction, assume $a. \in P''/M$ but $a. \notin P'/M$. Let $r \in P$ denote some rule with $head(r) = a$ which has survived the reduct modifications in P''/M. That is, each negated literal in $body(r)$ is false, i.e., $neg(r) \cap M = \emptyset$, so have again the condition from (8) as before. Now since $a. \notin P'/M$ we have either (1) r is not contained in P but $r' = a \leftarrow neg(r) \setminus \{\texttt{not } a\}$ is unacceptable in P or (2) r is a self-attacker which has been added when building the kernel of P. In any other cases, r would be contained in P as well. In both cases, $\texttt{not } a \in body(r)$ implies $M \cap neg(r) \neq \emptyset$; this is a contradiction to r witnessing $a. \in P''/M$.

We obtain $P'/M = P''/M$ for each stable model M of P' which implies that M is a stable model of P'' as well.

($\Leftarrow$) For the other direction, consider a stable model M of P'', i.e., let

$$M \in stb(P^{sk} \cup H) = stb(P'').$$

We show again that $P'/M = P''/M$. For each rule in the reduct obtained from some rule $r \in H$, the statement holds true. Moreover notice that all other rules not originating from rules in H are facts. In case $a. \in P''/M$ but $a. \notin P'/M$ we proceed as above (notice that we did not make use of the fact that M was a model of P' and not of P'').

For the other direction, let us assume $a. \in P'/M$ but $a. \notin P''/M$. Let $r \in P$ denote some rule with $head(r) = a$ which has survived the reduct modifications. That is, each negated literal in $body(r)$ is false, i.e., our condition (8) stating that $neg(r) \cap M = \emptyset$ holds.

Again, we distinguish the cases (1) r is deleted when building the kernel of P or (2) r is strongly unacceptable in P and thus the modified rule $r' = a \leftarrow neg(r) \cup \{\texttt{not } a\}$ is deleted when building the reduct of $P^{sk} \cup H$.

- In case 2 we let $t \in P$ be a rule witnessing unacceptability of r in P which is minimal in this aspect, i.e., $\{head(t)\} \cup neg(t)$ is $\subseteq$-minimal among all such rules. Then it holds that $t' = head(t) \leftarrow body(t) \cup \{\texttt{not } head(t)\}$ is contained in P^{sk}. Moreover, $neg(t') \subseteq neg(r)$. Since M is a model of P'' we obtain that $neg(t') \cap M \neq \emptyset$ (otherwise, it holds that $head(t) \in M$ and $head(t) \notin M$ by definition of stable model semantics). Hence $neg(r) \cap M \neq \emptyset$, contradiction to our assumption.

- For case 1 we perform only syntactical modifications, that is, we can proceed analogously to case 1 for the other direction. This concludes the proof of the statement. □

By our above results, we obtain that strong equivalence w.r.t. stable semantics coincides in the class of atomic and normal LPs when we compare atomic LPs.

Theorem 8.17. *$P \equiv^{stb} Q$ in the class of atomic LPs iff $P \equiv^{stb} Q$ in the class of normal LPs for any two atomic LPs P and Q.*

Proof. In case P and Q are not strongly equivalent in the class of atomic LPs we obtain that they are not strongly equivalent in the class of normal LPs as the former is a special case. Now assume P and Q are strongly equivalent in the class of atomic LPs. Then their stable kernels coincide (by Theorem 7.38). By Proposition 8.16, we obtain $P \equiv^{stb} P^{sk} = Q^{sk} \equiv^{stb} Q$ in the class of normal LPs. □

Corollary 8.18. *Deciding whether two atomic LPs P and Q are strongly equivalent w.r.t. stable semantics in the class of normal LPs is tractable.*

9. Discussion

Let us conclude by summarizing the present paper as well as discussing related and conceivable future work directions.

9.1 Summary & Technical Take-Aways

In this paper, we investigated two of the most important dynamic reasoning tasks, namely *strong equivalence* and *enforcement* within the context of assumption-based argumentation. We showed that in general, a very basic notion of enforcement (Theorem 4.5) as well as the natural notion of strong equivalence (Theorem 4.8) are intractable in ABA. Since the corresponding problems are known to be tractable within the realm of abstract argumentation (see Theorems 3.3 and 3.8), we identified two reasons for the mismatch in the computational complexity: i) the relation between the arguments in the instantiated AF and the rules in the knowledge base is too loose; ii) usual AFs are tailored for a static translation from a knowledge base into an AF, but not suitable for dynamic scenarios. Regarding i), we identified *atomic ABAs* (see Definition 5.14) as a promising, but nonetheless expressive ABA fragment. However, regarding ii) we had to readjust the instantiation procedure for our purpose. It turned out that the usual instantiation via AFs as proposed by Dung (Dung, 1995) abstracts away too much information for this endeavor, because one cannot know if and how an argument can be attacked in future scenarios since the AF only reflects the current, static situation. We therefore proposed claim and vulnerability augmented AFs (cvAFs; see Definition 5.2) which carry enough information to anticipate the role of an argument after the underlying knowledge base is updated.

Regarding the *enforcement* problem we observed that indeed, the most basic notions are tractable for cvAFs (Theorem 6.9, Proposition 6.12, and Corollary 6.13). Investigating variations of the enforcement notion showed that there is a gap in computational complexity in enforcing a set of arguments vs. a set of conclusions (Theorem 6.17 vs. Theorem 6.19), except for grounded semantics where both are intractable (Theorem 6.20). In a version where we want to enforce an argument without introducing a given set of claims in order to achieve this goal, the problem is intractable for any semantics (Theorem 6.23).

We showed that *strong equivalence* for two cvAFs $\mathcal{F}$ and $\mathcal{G}$ can be decided similar in spirit to the analogous problem for Dung-AFs by calculating semantics-dependent *kernels*. We want to emphasize that for all of our kernels, i.e., the complete in Definition 7.10, preferred in Definition 7.25, grounded in Definition 7.29, as well as stable in Definition 7.37, we mostly focus on modifications of the arguments rather than the attack relation (which is the case for abstract AFs). Since it suffices to compare the kernels of two cvAFs in order to decide strong equivalence (Theorems 7.20, 7.26, 7.30, and 7.38), this problem is tractable.

Finally, we *translated* the results into the realm of ABA. Interestingly, the two aforementioned problems are still intractable for atomic ABAs (Corollaries 6.24 and 7.40) and we had to additionally impose *separated contraries* (Definition 6.25). It was then possible to apply our cvAFs to obtain tractability for the enforcement problem (Theorem 6.26) and strong equivalence (Theorem 7.43) within this ABA fragment.

Take-Away 9.1. *The tractable ABA fragment consists of* atomic, flat ABA frameworks with separated contraries*. Each such ABA framework satisfies the following conditions:*

(i) each rule corresponds to an argument in the instantiated framework, and

(ii) the attacking elements (claims) and the defeasible elements (assumptions) of all arguments are strictly separated.

Point (i) is guaranteed by the restriction to atomic frameworks which prevents the chaining of rules. Each ABA framework can be transformed into an atomic one; as shown in (Rapberger, Ulbricht, & Wallner, 2022) this can even be achieved in polynomial time while preserving the semantics under projection.

The restriction to flat ABA frameworks with separated contraries guarantees point (ii). In flat frameworks, assumptions cannot be derived. Together with the condition that the assumptions and their contraries are separated, this implies that all attacks are undermining attacks that arise from the claim (a non-assumption) and target an element in the body of the attacked argument (an assumption). Each ABA framework can be translated into a ABA framework with separated contraries, as discussed in Section 6.4.

Emphasizing the generality of our approach using cvAFs, we applied our techniques to LPs as well. The high level point of view is that for LPs, the behavior is rather similar: Both enforcement (Proposition 8.12) and strong equivalence (known from the literature) are intractable in the general case, but become tractable for atomic LPs, as we can show by applying our cvAF results (Theorems 8.13 and 8.15). The LP results can be even entailed more straightforwardly, since the additional handling of the assumptions is not necessary.

Take-Away 9.2. *The tractable LP fragment consists of* atomic LPs*. Similar as for ABA frameworks, each rule corresponds to an argument in the instantiated framework. Moreover, the sets of attacking elements (atoms) and defeasible elements (default negated atoms) are disjoint in LPs by definition.*

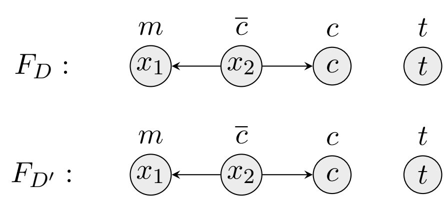
Notably, we could show that for stable semantics, deciding strong equivalence for two atomic LPs is tractable even if we allow arbitrary expansions (Theorem 8.18).

Along the way (and independent of our cvAF techniques) we characterized strong equivalence in ABA under stable semantics by means of SE-models, similar in spirit to analogous research for LPs.

9.2 Related Work

Our cvAFs generalize Dung-style AFs with a particular focus on instantiation-based settings. Apart from several notable generalizations of AFs that have been considered in the literature, e.g., AFs with collective attacks (Nielsen & Parsons, 2007), preference-based AFs (Kaci, van der Torre, Vesic, & Villata, 2021), or incomplete AFs (Baumeister, Neugebauer, Rothe, & Schadrack, 2018; Baumeister, Järvisalo, Neugebauer, Niskanen, & Rothe, 2021), we mention a particular generalization that is closely related to our cvAFs, called *claim-augmented argumentation frameworks (CAFs)* (Dvořák & Woltran, 2020). CAFs generalize AFs by assigning a claim to each argument and are therefore well suited to investigate instantiation-based settings; they are closely related to ABA frameworks and logic programs (Rapberger, 2020; König et al., 2022). Their properties in terms of expressiveness and computational complexity is well-studied (Dvořák et al., 2020; Dvořák, Greßler, Rapberger, & Woltran, 2023). However, when considering them in a dynamic setting, we observe similar issues as with AFs.

Example 9.3. *Let us consider our ABA frameworks D and D' from our introductory example (cf. Example 1.1) once again. This time, we instantiate both frameworks as CAF by keeping track of the claims of the arguments:*



Both CAFs F_D and $F_{D'}$ are identical. This is fine when considering static scenarios, however, when moving from static to dynamic settings, we experience the same issues as with AFs: we cannot identify differences between the two frameworks by looking at the abstract representation only.

Our cvAFs go one step further as CAFs and keep also track of the vulnerabilities of the arguments. By doing so, we are able to distinguish our running examples D and D' on the abstract level. Hence our generalization captures changes in a natural way.

A notion similar to the vulnerabilities of an argument has recently been studied (Prakken, 2022). In this work, Prakken investigates the dialectical strength of an argument in terms of possible attacks in framework expansions. In contrast to CAFs where each argument is labeled with its own claim, he equips each argument with a set of so-called *attack points* which correspond to its vulnerabilities, i.e., to the defeasible elements in the structure of the argument. Since we study enforcement and strong equivalence in terms of the conclusion-based outcome of ABA frameworks and LPs, it is crucial in our setting to consider not only the vulnerabilities but also the claim on the abstract level.

Dynamics. Our work extends research on dynamics in argumentation (Rotstein, Moguillansky, García, & Simari, 2010; Rotstein, Moguillansky, Falappa, García, & Simari, 2008; Snaith & Reed, 2017; Ulbricht & Baumann, 2019; Baumann, 2012a; Odekerken, Bex, Borg, & Testerink, 2022). In the context of AFs, both enforcement and strong equivalence are well-studied (Oikarinen & Woltran, 2011; Wallner et al., 2017; Baumann, 2012a).

Our characterization results for *strong equivalence* are similar to existing studies for other abstract representations such as AFs (Oikarinen & Woltran, 2011), CAFs (Baumann et al., 2022), or AFs with collective attacks (Dvorák, Fandinno, & Woltran, 2019). As for cvAFs, deciding strong equivalence can be characterized via semantics-dependent kernels. However, in contrast to the aforementioned abstract formalisms, the kernels for cvAFs are constructed by removing and modifying arguments. In logic-based approaches, a similar behavior has been observed: it is shown (Amgoud, Besnard, & Vesic, 2014) that under certain conditions on the underlying logic, unnecessary arguments can be removed while retaining (strong) equivalence. Although our work focuses on assumption-based argumentation, our cvAFs are constructed in a way such that they are independent of the underlying formalism, making them applicable in a more general setting. Our strong equivalence characterizations for ABA in terms of SE-models are inspired by similar characterizations for logic programs (Turner, 2001), simplifying the characterization via the Logic of Here and There (Lifschitz et al., 2001). In this regard, we furthermore mention Baumann and Strass (Baumann & Strass, 2022) who provide logic-based characterization results of strong equivalence in non-monotonic knowledge representation formalisms in a similar spirit. Moreover, strong equivalence is similar in spirit to *stability* (Testerink, Odekerken, & Bex, 2019).

Enforcement has received much attention in the argumentation community in recent years (Wallner et al., 2017; Baumann, 2012a). Enforcement in abstract models is typically easy to characterize; often, research in this matter takes certain minimality criteria into account. In structured approaches, studies on enforcement have received increasing attention. In a recent paper (Borg & Bex, 2021) the authors study under which conditions in a structured argumentation formalism a given formula can be enforced. Similarly to our setting, another study considers situations where an AF undergoes certain changes, but the permitted modifications are constrained (Wallner, 2020). Similarly to our setting, Wallner's motivation are AFs instantiated from a knowledge base, specifically from assumption-based argumentation. He considers different types of constraints and dynamic operators, with focus on minimal changes of the knowledge base. Constraints on the possibly reachable expansions of a given cvAF are intrinsic to our approach. In contrast to those results (Wallner, 2020), we focus on establishing existence criteria and identifying tractable fragments. A further approach (Moguillansky, Rotstein, Falappa, García, & Simari, 2008) considers argumentative revision operators in the context of defeasible logic programming in order to warrant a desired conclusion. In contrast to our enforcement approach, their objective lies in revising a program such that an argument with the desired conclusion ends up undefeated.

More generally speaking, our work considers *changes* of knowledge bases, which is closely related to the area of belief revision (Alferes, Leite, Pereira, Przymusinska, & Przymusinski, 2000). Argumentation and belief revision are closely related (Falappa, Kern-Isberner, & Simari, 2009). Revising knowledge in argumentation has received some attention in recent years. Snaith and Reed (2017) study several revision operator with main focus on ASPIC$^+$;

Hadjisoteriou and Kakas (2015) develop a framework to express logic- based reasoning about actions and change; and Rotstein et al. (2008, 2010) develop a model to handle change in argumentation. In their model, they keep track of the structure of the arguments and their sub-argument relation at the abstract level; hence, we observe certain parallels to our cvAFs. In contrast to our approach, they consider both the addition and the removal of arguments and study associated interactions. Indeed, the main focus of our work are operations that are based on *framework expansions*. In this regard, we want to highlight in particular the work by Cayrol et al. (Cayrol, de Saint-Cyr, & Lagasquie-Schiex, 2010) who study framework expansions in the context of AFs. They consider several types of revision operators that impose certain properties of the outcome and establish conditions under which a given property is satisfied.

Redundancies. The redundancy notions we discussed are similar in spirit to the line of research on syntactic transformations for LPs (Brass & Dix, 1997; Eiter, Fink, Tompits, & Woltran, 2004; Wang & Zhou, 2005; Lin & Chen, 2007), that gave rise to alternative characterizations of strong equivalence (Osorio, Pérez, & Arrazola, 2001; Cabalar, 2002) and set the ground for further complexity analysis of LP fragments (Eiter, Fink, Tompits, & Woltran, 2007). We already have mentioned redundancy studies in logic-based argumentation (Amgoud et al., 2014). Redundancies have also been considered for CAFs (Dvořák et al., 2020) and AFs with collective attacks (Dvořák, Rapberger, & Woltran, 2020).

9.3 Future Work

Future work directions include exploring further formalisms where cvAFs are applicable, i.e., investigating suitability for e.g. ASPIC (Modgil & Prakken, 2018) or logic-based argumentation (Besnard & Hunter, 2001). As demonstrated in our LP section, utilizing cvAFs is a promising technique. Since ASPIC frameworks are not necessarily flat, i.e., the derivations are in general not constrained, it will be particularly interesting to identify the corresponding tractable fragment and to generalize our results to capture the considered dynamic tasks in ASPIC with our cvAFs. Similarly, finding more reasoning tasks where we can benefit from cvAFs would 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., ASPIC$^+$ (Modgil & Prakken, 2018) or ABA$^+$ (Cyras et al., 2018), do not always yield well-formed cvAFs when instantiating, so one could investigate whether it is possible to apply similar proof techniques to this setting.

As we have mentioned at the beginning of Section 6, our enforcement results for cvAFs rely on the assumption that each possible expansion is permitted. However, it can be the case that not every expansion is allowed (Prakken, 2022), for instance, because the underlying knowledge base imposes constraints on the argument construction or inclusion. Taking these considerations into account would be a challenging avenue for future research. More broadly, research on dynamics benefits from a solid understanding of the expressive power of the given formalism. This is well-studied in the the realm of abstract argumentation (Baumann & Strass, 2013; Dunne, Dvořák, Linsbichler, & Woltran, 2015; Ulbricht, 2021), however, for ABA this line of research has mostly been neglected so far.

We extended the research on SE-models from LPs to ABA, but only for stable semantics. The crucial feature of stable semantics (both in ABA and LPs) is that the models are two-valued. It is not clear how to characterize strong equivalence for three-valued semantics in a similar way. We are convinced this is an interesting avenue for future research.

We furthermore note that studies related to our redundancy notions in the context of structured approaches or logic programs could be an interesting avenue for future research. As demonstrated in Section 8, our redundancy notions are easily transferable to atomic instances. It would be interesting to generalize these notions to general LPs or ABA frameworks.

Acknowledgements

We all thank reviewers of previous versions of this paper for their thorough feedback and helpful comments which significantly contributed to the current version of this work. This research has been supported by Vienna Science and Technology Fund (WWTF) through project ICT19-065, the Austrian Science Fund (FWF) through projects P32830 and W1255-N23 as well as 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.

Appendix A. Computational Complexity of Dynamic Tasks for ABA

Theorem 4.5. *Deciding whether a conclusion p (assumption a) is enforceable in a given ABA framework D w.r.t. a semantics $\sigma \in \{gr, co, pr, stb\}$ is* NP*-hard.*

Proof. We present a reduction from SAT which shows hardness for grounded, complete, preferred, and stable semantics. Given a CNF formula φ with clauses $C = \{c_1, \dots, c_n\}$ over variables in X, we let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be defined as in Reduction 4.4.

We show φ is enforceable w.r.t. σ iff φ is satisfiable.

First assume φ is satisfiable and let $M \subseteq X$ be a model of φ. For each $x \in M$, we introduce rules of the form "$\overline{x_a^T} \leftarrow$" and for each $x \notin M$, we add rules "$\overline{x_a^F} \leftarrow$". Each of these conclusions is contained in the grounded extension (is derivable by the empty set of assumptions $\mathcal{E}$). Moreover, for each $x \in X$, if $\mathcal{E} \vdash \overline{x_a^T}$ then x_a^F is unattacked and thus contained in the grounded extension G (since we have introduced a fact precisely for each atom). G contains the assumptions c and e: Since M is a satisfying assignment of φ, each clause-rule with head $\overline{c}$ is attacked by the newly introduced rules, thus we have $c \in G$. Moreover, for every $x \in X$, either x_p^T or x_p^F is attacked by G, thus $e \in G$. We obtain $G \vdash \varphi$.

We observe that the AF arising from D is acyclic (clearly, also after adding facts to D), thus $gr(D) = co(D) = pr(D) = stb(D)$. Consequently, φ is satisfiable implies the conclusion φ is enforceable under all considered semantics.

Now assume φ is unsatisfiable. Towards a contradiction, assume φ is enforceable w.r.t. σ. That is, there is a set of rules $\mathcal{R}'$, there is a σ-assumption-set $A \subseteq \mathcal{A}$, such that φ is derivable by A in $D' = (\mathcal{L}, \mathcal{R} \cup \mathcal{R}', \mathcal{A}, ^-)$. This is the case if A defends φ against all attacks. Consequently, (a) for each $x \in X$, $\mathcal{R}'$ contains either rules with conclusion $\overline{x_a^T}$ or $\overline{x_a^F}$ but not both, otherwise both x_a^T, x_a^F are not contained in G and thus the attack on e from $\{x_p^T, x_p^F\}$ stays undefeated; also, (b) for each $i \leq n$, $\mathcal{R}'$ contains some rule with conclusion $\overline{a}$ for some

$a \in A_i$, that is, either $\overline{x_a^T}$ or $\overline{x_a^F}$ for some $x \in X$. Thus for all c_i, either $G \vdash \overline{x_a^T}$ in case $x \in c_i$ or $G \vdash \overline{x_a^F}$ in case $\neg x \in c_i$. We obtain that $M = \{x \mid G \vdash x_a^T\}$ is a satisfying assignment of φ, contradiction to the assumption φ is unsatisfiable.

To show NP-hardness of assumption-enforcement, we adapt Reduction 4.4 as follows: For a CNF formula φ with clauses $C = \{c_1, \dots, c_n\}$ over variables in X, we define the corresponding ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ with

- $\mathcal{A} = \{x_a^T, x_p^F, x_a^F, x_p^T \mid x \in X\} \cup \{\varphi\}$;
- $\overline{x_p^F} = x_a^T$, $\overline{x_p^T} = x_a^F$, and $\overline{x_a^T}, \overline{x_a^F}, \overline{\varphi} \in \mathcal{L} \setminus \mathcal{A}$.

Moreover, $\mathcal{R}$ contains the following rules:

- for all $x \in X$, $\mathcal{R}$ contains a rule $\overline{\varphi} \leftarrow x_p^T, x_p^F$;
- for each $i \leq n$, $\mathcal{R}$ contains a rule of the form $\overline{\varphi} \leftarrow \{x_a^T \mid x \in c_i\} \cup \{x_a^F \mid \neg x \in c_i\}$.

Considering the example in Figure 1, we have replaced all arguments concluding $\overline{e}$ or $\overline{c}$ with arguments concluding $\overline{\varphi}$ without changing the incoming attacks. The remaining part of the proof is analogously to the proof for conclusion-enforcement as outlined above. □

Theorem 4.8. *Deciding whether two ABA frameworks are strongly equivalent w.r.t. a given semantics $\sigma \in \{gr, co, pr, stb\}$ is* coNP*-hard.*

Proof. We present a reduction from UNSAT: Given a CNF formula φ with clauses $C = \{c_1, \dots, c_n\}$ over variables in X, we let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be defined as in Reduction 4.4, and $D' = (\mathcal{L}, \mathcal{R}', \mathcal{A}, ^-)$ with $\mathcal{R}' = \mathcal{R} \setminus \{\varphi \leftarrow c, e\}$, that is, we consider two independent frameworks that differ in a single rule: D' has no argument for φ. If some expansion of D' has a σ-assumption-extension concluding φ this is only because an argument with conclusion φ has been added when expanding D'. By our results from Theorem 4.5, we have that φ is satisfiable iff there is a set of rules $\mathcal{R}''$ such that $(\mathcal{L}, \mathcal{R} \cup \mathcal{R}'', \mathcal{A}, ^-)$ admits a σ-assumption-extension that concludes φ. Consequently, φ is satisfiable iff there is some expansion D'' of D and D' such that $\sigma(D \cup D'') \neq \sigma(D' \cup D'')$, i.e., D and D' are not strongly equivalent to each other. □

Appendix B. Computational Complexity of Dynamic Tasks for cvAFs

Theorem 6.20. *Deciding whether a set X of arguments is gr-enforceable for a given cvAF $\mathcal{F} = (A, R)$ is* NP*-hard.*

Sketch of proof. Consider the following reduction.

Reduction B.1. *For a CNF formula φ with clauses $C = \{c_1, \dots, c_n\}$ over variables in $X = \{x_1, \dots, c_m\}$, we define the corresponding cvAF $\mathcal{F} = (A, R)$ with*

$$A = C \cup \{v_1, \dots, v_m\} \cup \{\bar{v}_1, \dots, \bar{v}_m\} \cup \{U_1, \dots, U_m\} \cup \{p_1, \dots, p_m\}$$

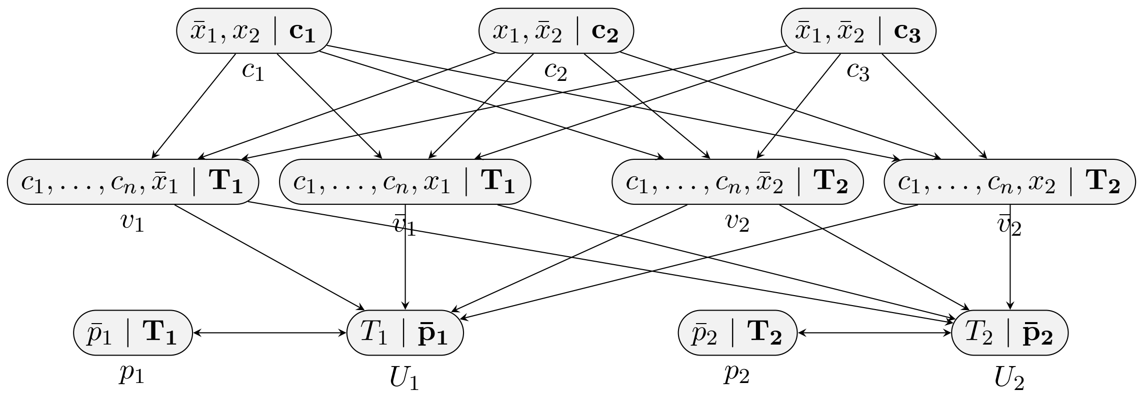


Figure 6: Reduction B.1 applied to the formula ϕ consisting of clauses $c_1 = \{\neg x_1, x_2\}$, $c_2 = \{x_1, \neg x_2\}$, $c_3 = \{\neg x_1, \neg x_2\}$

where

$$
\begin{aligned}
&\forall c \in C : cl(c) = c && vul(c) = \{x_j \mid x_j \in c\} \cup \{\bar{x}_j \mid \neg x_j \in c\}\\
&\forall 1 \leq i \leq m : cl(v_i) = T_i && vul(v_i) = C \cup \{\bar{x}_i\}\\
&\forall 1 \leq i \leq m : cl(\bar{v}_i) = T_i && vul(\bar{v}_i) = C \cup \{x_i\}\\
&\forall 1 \leq i \leq m : cl(U_i) = \bar{p}_i && vul(U_i) = T_i\\
&\forall 1 \leq i \leq m : cl(p_i) = T_i && vul(p_i) = \{\bar{p}_i\}
\end{aligned}
$$

and the induced attack relation. An example of this reduction can be found in Figure 6.

The reduction is an extension of Reduction 6.18 (see also the proof of Theorem 6.19). Given a formula ϕ in CNF we construct $\mathcal{F} = (A, R)$ as described in Reduction B.1. We claim that ϕ is satisfiable iff $\{p_1, \ldots, p_m\}$ can be enforced in $\mathcal{F}$. For this, recall the proof of Theorem 6.19: $\{T_1, \ldots, T_m\}$ can be enforced iff ϕ is satisfiable. However, since we cannot introduce $\{T_1, \ldots, T_m\}$ directly, in order to ensure $p_1, \ldots, p_m$ occurs in the grounded extension, we have to defeat $U_1, \ldots, U_m$ by proceeding as in the proof of Theorem 6.19. $\square$

Theorem 6.23. *Deciding whether a set X of arguments (a set C of claims) is D-eluding σ-enforceable for a given cvAF $\mathcal{F} = (A, R)$ and semantics $\sigma \in \{co, gr, pr, stb\}$ is* NP*-hard.*

Proof. For claim-enforceability and for $\sigma = gr$, we obtain the result from Corollary 6.22. Hence let us focus on argument enforcement for the semantics $\sigma \in \{co, pr, stb\}$. For a CNF formula φ with clauses C over variables in Y, we let $\bar{Y} = \{\bar{y} \mid y \in Y\}$. We construct arguments corresponding to the clauses as follows:

$$C_A = \{(Z, c) \mid c \in C, Z = \{y \mid y \in c \cap Y\} \cup \{\bar{y} \mid \neg y \in c \cap Y\}\}.$$

We define the corresponding cvAF $\mathcal{F} = (A, R)$ with

$$A = (Y \times \bar{Y}) \cup (\bar{Y} \times Y) \cup C_A \cup \{(C, \varphi)\}$$

and the induced attack relation R. Let $a = (C, \varphi)$. The cvAF corresponds to the standard reduction (Dvořák & Dunne, 2018, Reduction 3.6). As discussed in (Dvořák & Dunne, 2018), φ is satisfiable iff the argument a is credulously accepted. We will make use of this result for our enforcement scenario as follows:

We set $X = \{a\}$ and $D = Y \cup \bar{Y} \cup \{\varphi\}$. That is, we aim to enforce the argument a (corresponding to the formula φ) and we are not allowed to introduce any claim corresponding to literals or φ. Hence we cannot defend φ against the attacks from the clause arguments C (to do so, we would need to introduce claims from $Y \cup \bar{Y}$). Therefore, it holds that a is D-eluding σ-enforceable iff a is already credulously accepted in $\mathcal{F}$. From the results for the standard reduction we discussed above we obtain that a is D-eluding σ-enforceable iff φ is satisfiable. This proves our desired lower bounds. □

Appendix C. Strong Equivalence

We present the omitted proofs for preferred, grounded, and stable semantics.

C.1 Preferred Kernel

Proposition 7.24. *For a cvAF $\mathcal{F} = (A, R)$ and a strongly pr-unacceptable argument $x \in A$, $pr_{cl}(\mathcal{F}) = pr_{cl}(\mathcal{F} \setminus \{x\})$.*

Proof. Let $\mathcal{F}' = \mathcal{F} \setminus \{x\}$ and recall that x can never appear in an admissible extension as it it strongly unacceptable. Let x be strongly pr-unacceptable w.r.t. $y \in A$. Then $\Gamma_{\mathcal{F}}(\{z\}) \subseteq \Gamma_{\mathcal{F}'}(\{z\})$ for every $z \in A \setminus \{x\}$, i.e., every argument $z \neq x$ defends the same arguments in $\mathcal{F}'$ which are defended by z in $\mathcal{F}$. We obtain $ad_{cl}(\mathcal{F}) \subseteq ad_{cl}(\mathcal{F}')$.

To prove $pr_{cl}(\mathcal{F}) = pr_{cl}(\mathcal{F}')$ we show that for every $E \in ad(\mathcal{F}')$, there is $D \in ad(\mathcal{F})$ such that $E \subseteq D$. In case $E \in ad(\mathcal{F})$, we are done (taking $D = E$). In case $E \notin ad(\mathcal{F})$, there is $z \in E$ such that $(x, z) \in R$ and z is not defended by E in F. In case $E \cup \{y\} \in cf(\mathcal{F})$ we are done (note that in this case, $D = E \cup \{y\}$ is admissible). Now assume $E \cup \{y\}$ is not conflict-free. Observe that $(y, y) \notin R$ by assumption $vul(y) = \{cl(x)\}$. In case there is $v \in E$ such that $(v, y) \in R$ we have $cl(v) = c$ and thus $(v, x) \in R$ by well-formedness, contradiction to $E \notin ad(\mathcal{F})$. In case $(y, v) \in R$ for some $v \in E$ we have some $w \in E$ which defends v against w (since E is admissible in $\mathcal{F}'$) thus we arrive again at a contradiction since $(w, y) \in R$ implies $(w, x) \in R$. It follows that $D = E \cup \{y\}$ is an admissible superset of E in $\mathcal{F}$. We have shown that the preferred extensions of $\mathcal{F}$ and $\mathcal{F}'$ coincide. □

To show Theorem 7.26 we proceed similarly to the case of complete semantics. First, we show that each cvAF is strongly equivalent to its preferred kernel.

Proposition C.1. $\mathcal{F} \equiv_s^{pr} \mathcal{F}^{ck}$ *for every cvAF $\mathcal{F}$.*

Proof. Consider a set X of instantiated arguments. By Proposition 7.12, we have that $pr_{cl}(f_e(\mathcal{F}^{ck}, X)) = pr_{cl}(f_e(\mathcal{F}, X))$. Let Let $A_{punac} \subseteq A^{ck}$ denote the set of strongly pr-unacceptable arguments of $\mathcal{F}^{ck}$. By Proposition 7.24, we can delete strongly pr-unacceptable arguments iteratively without changing preferred extensions. We obtain $pr_{cl}(f_e(\mathcal{F}', X)) = pr_{cl}(f_e(\mathcal{F}, X))$ for $\mathcal{F}' = \mathcal{F} \setminus A_{punac}$. By definition of the preferred kernel, it holds that $\mathcal{F}' = \mathcal{F}^{pk}$. Hence we obtain $\mathcal{F} \equiv_s^{pr} \mathcal{F}^{pk}$. □

As a corollary, we obtain that preferred semantics of each cvAF and its preferred kernel coincides.

Corollary C.2. *$pr_{cl}(\mathcal{F}) = pr_{cl}(\mathcal{F}^{ck})$ for every cvAF $\mathcal{F}$.*

Next, we show that the preferred kernel does not contain redundant, strongly defeated, and strongly pr-unacceptable arguments; moreover, each strongly unacceptable argument is self-attacking. While the latter follows by Lemma 7.15, it remains to show that redundant, strongly defeated, and strongly pr-unacceptable arguments can be removed iteratively.

Lemma C.3. *Given a cvAF $\mathcal{F}$ and arguments $x, y \in A$, $x \neq y$. Let y be redundant, strongly defeated, or strongly pr-unacceptable in $\mathcal{F}$. Then x is redundant, strongly defeated, or strongly pr-unacceptable in $\mathcal{F}$ iff x is redundant, strongly defeated, or strongly pr-unacceptable in $\mathcal{F} \setminus \{y\}$.*

Proof. We first observe that if x is redundant, strongly defeated, or strongly pr-unacceptable in $\mathcal{F} \setminus \{y\}$ then there is a witness z in $\mathcal{F} \setminus \{y\}$. As mentioned in Observation 7.14, the claim-attacks are not affected by removing certain arguments. We thus obtain that z witnesses that x is redundant, strongly defeated, or strongly pr-unacceptable in $\mathcal{F}$. Also, in case x is strongly defeated in $\mathcal{F}$, it is clear that x is contained in $\mathcal{F} \setminus \{y\}$ since y is not unattacked and thus cannot witness that x is is strongly defeated.

Let y be strongly defeated in $\mathcal{F}$. In case x is redundant w.r.t. y in $\mathcal{F}$, there is some unattacked $z \in A$ with $(z, y) \in R$. Thus we obtain that also x is strongly defeated (using $vul(y) \subseteq vul(x)$, i.e., $(z, x) \in R$). In case x is strongly pr-unacceptable w.r.t. y in $\mathcal{F}$, there is some unattacked $z \in A$, $(z, y) \in R$, moreover, $cl(z) = cl(x)$ (using $vul(y) = \{cl(x)\}$. Consequently we obtain that x is redundant in $\mathcal{F}$ and in $\mathcal{F} \setminus \{y\}$.

Let y be redundant in $\mathcal{F}$ and let x be redundant w.r.t. y in $\mathcal{F}$. Then there is $z \in A$ with $vul(z) \subseteq vul(y)$ and $cl(z) = cl(y)$, thus witnessing the redundancy of x. In case x is strongly pr-unacceptable w.r.t. y in $\mathcal{F}$, there is $z \in A$ with $cl(z) = cl(y)$ and $vul(z) \subset vul(y) = \{cl(x)\}$, thus $vul(z) = \emptyset$; moreover, $(z, x) \in R$ using $cl(z) = cl(y) \in vul(x)$. We obtain that x is strongly defeated. □

We remark that the lemma states that the disjunction of the three properties is preserved. Similar as in Lemma 7.17, a redundant argument can turn into a strongly defeated argument when removing y.

Similarly as in the case of the complete kernel, we obtain that computing the preferred kernel of a given cvAF does not 'forget' any redundancies. By Lemma 7.15, Observation 7.14, and Lemma C.3, we obtain the following result, which is proven analogously to Proposition 7.18.

Proposition C.4. *For any cvAF $\mathcal{F}$, the kernel $\mathcal{F}^{pk}$ does neither contain redundant, non-self-attacking strongly unacceptable, strongly defeated or strongly pr-unacceptable arguments.*

Again, the next auxiliary result is to establish that preferred kernels of two strongly equivalent cvAFs contain the same claims.

Lemma C.5. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{pr} \mathcal{G}$ implies $cl(A_{\mathcal{F}^{pk}}) = cl(A_{\mathcal{G}^{pk}})$.*

Proof. Let $x \in A_{\mathcal{F}^{pk}}$ with $cl(x) = c$. Towards a contradiction, assume that there is no argument $y \in A_{\mathcal{G}^{pk}}$ with $cl(y) = c$. Since we may assume $pr_{cl}(\mathcal{F}^{pk}) = pr_{cl}(\mathcal{G}^{pk})$ in this case we deduce that x does not occur in any preferred extension of $\mathcal{F}^{pk}$. Hence it does not occur in any admissible extension. Consequently, x receives incoming attacks.

We proceed similarly as in the proof of Lemma 7.19.

Case 1 First, we suppose that x is no self-attacker. This case is analogously to the proof of Lemma 7.19.

Case 2 Now suppose each argument with claim c is a self-attacker and fix such x. Since x occurs in the kernel $\mathcal{F}^{pk}$, each attacker of x must itself possess attacking arguments. This case is analogously to Case 2.1 in the proof of Lemma 7.19. Since the preferred kernel does not contain pr-unacceptable arguments, it holds that each attacker z of x contains some vulnerability $e \in vul(z)$ with $e \neq c$. Hence a case analogously to Case 2.2 in the proof of Lemma 7.19 can never occur. □

We are ready to prove our main result of this section.

Theorem 7.26. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{pr} \mathcal{G}$ iff $\mathcal{F}^{pk} = \mathcal{G}^{pk}$.*

Proof. To prove the statement, we proceed analogously to the proof of Theorem 7.20 for complete semantics.

First assume $\mathcal{F}^{pk} = \mathcal{G}^{pk}$ holds. By Proposition C.1, it holds that $\mathcal{F}^{pk} \equiv_s^{pr} \mathcal{F}$ and $\mathcal{G}^{pk} \equiv_s^{pr} \mathcal{G}$. Thus we obtain $\mathcal{F} \equiv_s^{pr} \mathcal{G}$ by transitivity.

For the other direction, assume $\mathcal{F} \equiv_s^{pr} \mathcal{G}$. By Lemma C.5, it holds that $\mathcal{F}^{pk}$ and $\mathcal{G}^{pk}$ contain the same claims. Analogously to the proof of Theorem 7.20, it can be shown that for all arguments x in $\mathcal{F}^{pk}$ there is some argument y in $\mathcal{G}^{pk}$ such that $cl(x) = cl(y) = c$ and $vul(y) \subseteq vul(x)$. Hence we conclude that $\mathcal{F}^{pk}$ and $\mathcal{G}^{pk}$ contain the same arguments. We obtain that $\mathcal{F}^{pk} = \mathcal{G}^{pk}$. □

C.2 Grounded Kernel

Proposition 7.28. *Given a cvAF $\mathcal{F} = (A, R)$ and a strongly gr-unacceptable argument $x \in A$ and let $x' = (vul(x) \cup \{cl(x)\}, cl(x))$. Then $gr(\mathcal{F}) = gr((f_e(\mathcal{F} \setminus \{x\}, x'))$.*

Proof. Let $\mathcal{F}' = f_e(\mathcal{F} \setminus \{x\}, x')$ and assume x is strongly gr-unacceptable w.r.t. $y \in A$. In case $x \notin gr(\mathcal{F})$ we are done (turning x into a self-attacking argument does not change the grounded extension). In case $x \in gr(\mathcal{F})$ there is $z \in gr(\mathcal{F})$ such that $(z, y) \in R$. If $cl(z) \neq cl(x)$ we have $cl(z) \in vul(x)$ by assumption $vul(y) \setminus \{cl(x)\} \subseteq vul(x)$, that is, z attacks x, contradiction to $\{x, z\} \subseteq gr(\mathcal{F})$. In case $cl(z) = cl(x)$, we have $cl(x) \in gr_{cl}(\mathcal{F}')$, and z attacks the same arguments as x by well-formedness, hence $gr_{cl}(\mathcal{F}) = gr_{cl}(\mathcal{F}')$. □

Let us now infer the usual auxiliary results leading to Theorem 7.30.

Proposition C.6. *$\mathcal{F} \equiv_s^{gr} \mathcal{F}^{ck}$ for every cvAF $\mathcal{F}$.*

Proof. By Proposition 7.28, we can modify all strongly gr-unacceptable arguments of $\mathcal{F}$ without changing semantics. Next, we iteratively remove all redundant and strong unacceptable arguments (cf. Proposition 7.5 and 7.6). □

Corollary C.7. *$gr_{cl}(\mathcal{F}) = gr_{cl}(\mathcal{F}^{ck})$ for every cvAF $\mathcal{F}$.*

Lemma C.8. *For a cvAF $\mathcal{F} = (A, R)$ and a strongly gr-unacceptable argument $x \in A$. Let $x' = (vul(x) \cup \{cl(x)\}, cl(x))$ and let $\mathcal{F}' = f_e(\mathcal{F} \setminus \{x\}, x') = (A', R')$. Then, for all $y \neq x \in A$, y is strongly gr-unacceptable in $\mathcal{F}$ iff y is strongly gr-unacceptable in $\mathcal{F}'$.*

Proof. Let $y \in A$ be strongly gr-unacceptable in $\mathcal{F}$. Then there is $z \in A$ with $vul(z) \setminus \{cl(y)\} \subseteq vul(y)$ and $(z, y) \in R$ in $\mathcal{F}$. In case $z \neq x$ we are done (then $z \in A'$). In case $z = x$, we have $cl(x) \in vul(y)$. Replacing x in $\mathcal{F}'$ with x', we obtain $vul(x') = vul(x) \cup \{cl(x)\}$, thus $vul(x') \setminus \{cl(y)\} \subseteq vul(y)$ and $(x', y) \in R$ showing that y is strongly (gr-)unacceptable in $\mathcal{F}'$. In case $y \in A'$ is strongly gr-unacceptable in $\mathcal{F}'$, there is a witness $z \in A'$ in $\mathcal{F}$. Recall that $A' \setminus \{x\} \subset A \setminus \{x\}$. We proceed by case distinction: First assume $z \neq x'$. Then $z \in A$ and thus we obtain that y is strongly gr-unacceptable in $\mathcal{F}$. Now assume $z = x'$. Then x attacks y in $\mathcal{F}$ and $vul(x) \setminus \{cl(x)\} = vul(x') \setminus \{cl(x)\} \subseteq vul(y)$. This shows that y is strongly gr-unacceptable in $\mathcal{F}$. □

Analogously to Proposition 7.18, we obtain the following result.

Proposition C.9. *The grounded kernel $\mathcal{F}^{ck}$ of a cvAF $\mathcal{F}$ does not contain redundant nor strongly defeated arguments, and each strongly gr-unacceptable argument is self-attacking.*

Similarly as for complete and preferred semantics, we show that the grounded kernels of two strongly equivalent cvAFs contain the same claims.

Lemma C.10. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{gr} \mathcal{G}$ implies $cl(A_{\mathcal{F}^{gk}}) = cl(A_{\mathcal{G}^{gk}})$.*

Proof. Let $x \in A_{\mathcal{F}^{gk}}$ with $cl(x) = c$. Towards a contradiction, assume that there is no argument $y \in A_{\mathcal{G}^{gk}}$ with $cl(y) = c$. Since we may assume $gr_{cl}(\mathcal{F}^{gk}) = gr_{cl}(\mathcal{G}^{gk})$ in this case we deduce that x does not occur in the grounded extension of $\mathcal{F}^{gk}$. Consequently, x receives incoming attacks.

Case 1 Suppose x is no self-attacker. Consider the set $\mathcal{Z} = \{z \in A_{\mathcal{F}^{gk}} \mid (z, x) \in R_\mathcal{F}\}$ of arguments attacking x. Since x is no self-attacker, by definition of the kernel we have $vul(z) \setminus (vul(x) \cup \{cl(x)\}) \neq \emptyset$ for each $z \in \mathcal{Z}$. Thus by letting

$$V_\mathcal{Z} = \{v_e = (\emptyset, e) \mid e \in vul(z) \setminus (vul(x) \cup \{cl(x)\}), z \in \mathcal{Z}\},$$

we defeat these attackers without introducing claim c. Thus c appears in the grounded extension of $f_e(\mathcal{F}^{gk}, V_\mathcal{Z})$ but not in $f_e(\mathcal{G}^{gk}, V_\mathcal{Z})$.

Case 2 Now suppose each argument with claim c is a self-attacker and fix such x. Since x occurs in the kernel $\mathcal{F}^{gk}$, each attacker of x must itself possess attacking arguments.

First, we get rid of arguments with the same claim c. Let $\mathcal{Y} = \{y \in A_{\mathcal{F}^{gk}} \mid cl(y) = c, y \neq x\}$ denote the set of arguments with claim c. We consider arguments which defeat them; this time we can get rid of all of them via

$$\begin{aligned} V_\mathcal{Y} &= \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}, e \neq c\} \\ &= \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}\}. \end{aligned}$$

Now by introducing a self-attacker to each claim except c we ensure that all arguments except the unattacked ones are attacked and hence undecided in the unique grounded extension; in particular, x is. Thus consider $V = \{v_e = (\emptyset, e) \mid e \in cl(\mathcal{F}^{gk}), e \neq c\}$.

With the usual technique—introducing a fresh argument attacked by claim c—we separate the two cvAFs. □

We are ready to state the desired kernel characterization.

Theorem 7.30. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{gr} \mathcal{G}$ iff $\mathcal{F}^{gk} = \mathcal{G}^{gk}$.*

Proof. First assume $\mathcal{F}^{gk} = \mathcal{G}^{gk}$ holds. By Proposition 7.12, it holds that $\mathcal{F}^{gk} \equiv_s^{gr} \mathcal{F}$ and $\mathcal{G}^{gk} \equiv_s^{gr} \mathcal{G}$. Thus we obtain $\mathcal{F} \equiv_s^{gr} \mathcal{G}$ by transitivity.

For the other direction, assume $\mathcal{F} \equiv_s^{gr} \mathcal{G}$. By Lemma C.10, it holds that $\mathcal{F}^{gk}$ and $\mathcal{G}^{gk}$ contain the same claims. To show that for all arguments x in $\mathcal{F}^{gk}$ there is some argument y in $\mathcal{G}^{gk}$ such that $cl(x) = cl(y) = c$ and $vul(y) = vul(x)$, we proceed analogously to the proof of Theorem 7.20. Hence we obtain that $\mathcal{F}^{gk}$ and $\mathcal{G}^{gk}$ contain the same arguments, which yields $\mathcal{F}^{gk} = \mathcal{G}^{gk}$. □

C.3 Stable Kernel

Towards proving Theorem 7.38, we proceed by the usual steps. Iterative application of Proposition 7.33, 7.5, 7.7, and 7.6 shows that each cvAF is strongly equivalent to its stable kernel.

Proposition C.11. *$\mathcal{F} \equiv_s^{gr} \mathcal{F}^{sk}$ for every cvAF $\mathcal{F}$.*

Next we show that the deletion of strongly unacceptable, redundant, or strongly defeated arguments does not change strong unacceptability, redundancy, or strong defeat of other arguments. Hence such arguments can be iteratively removed.

Lemma C.12. *For a cvAF $\mathcal{F} = (A, R)$ and a strongly unacceptable, redundant, or strongly defeated argument $y \in \mathcal{F}$, $y \neq x \in A$ is strongly unacceptable, redundant, or strongly defeated in $\mathcal{F}$ iff x is strongly unacceptable, redundant, or strongly defeated in $\mathcal{F} \setminus \{y\}$.*

Proof. ($\Leftarrow$) We first observe that if x is strongly unacceptable, redundant, or strongly defeated in $\mathcal{F} \setminus \{y\}$ then there is a witness z in $\mathcal{F} \setminus \{y\}$. As mentioned in Observation 7.14, the claim-attacks are not affected by removing certain arguments. We thus obtain that z witnesses that x is strongly unacceptable, redundant, or strongly defeated in $\mathcal{F}$. Also, in case x is strongly defeated, it is clear that x is contained in $\mathcal{F} \setminus \{y\}$ since y is not unattacked and thus cannot serve as witness for x being strongly defeated.

($\Rightarrow$) To prove the other direction, we proceed by case distinction.

- Let y be strongly unacceptable.

  First, let x be strongly unacceptable in $\mathcal{F}$. In case y witnesses strong unacceptability of x in $\mathcal{F}$, there is z with $vul(z) \subseteq vul(y)$ and $cl(z) \in vul(y)$. Then z witnesses unacceptability of x in $\mathcal{F}$ since $vul(z) \subseteq vul(x)$ and $cl(z) \in vul(x)$. W.l.o.g., let z be minimal in the sense that there is no $u \in A$ with $vul(u) \subset vul(y)$ and $cl(u) \in vul(y)$. Then z is not strongly unacceptable in $\mathcal{F}$ (otherwise, we find such an u, contradiction to the minimality assumption), and thus z witnesses the unacceptability of x in $\mathcal{F} \setminus \{y\}$.

  In case x is redundant in $\mathcal{F}$ w.r.t. y, there is z with $vul(z) \subseteq vul(y)$ and $cl(z) \in vul(y)$. Thus $vul(z) \subseteq vul(x)$ and $cl(x) \in vul(y)$ shows that x is strongly unacceptable in $\mathcal{F}$. We obtain x is strongly unacceptable in $\mathcal{F} \setminus \{y\}$.

- Let y be strongly defeated. In case x is redundant w.r.t. y in $\mathcal{F}$, there is some $z \in A$ with $(z, y) \in R$. Thus we obtain that also x is strongly defeated (using $vul(y) \subseteq vul(x)$, i.e., $(z, x) \in R$). In case x is strongly unacceptable w.r.t. y in $\mathcal{F}$, also x is strongly defeated (using $vul(y) \subseteq vul(x)$).

- Let y be redundant. First, let x be redundant w.r.t. y. Then there is $vul(z) \subseteq vul(y)$ and $cl(z) = cl(y)$, thus witnessing the redundancy of x. In case x is unacceptable w.r.t. y in $\mathcal{F}$. There is $z \in A$ satisfying $vul(z) \subseteq vul(y)$ and $cl(z) = cl(y)$, and thus z witnesses unacceptability of x in $\mathcal{F} \setminus \{y\}$. □

We remark that the lemma states that the disjunction of the properties is preserved. Similar as in Lemma 7.17, a redundant argument can turn into a strongly defeated argument when removing y.

The stable kernel is constructed by (1) modifying all strongly unacceptable arguments; (2) adding all self-attacking arguments with conclusions $s \in vul(x)$ for each self-attacking argument x; and (3) removing all redundant, strictly unacceptable, and strongly defeated arguments; by Lemma C.12, we obtain the following.

Proposition C.13. *For a cvAF $\mathcal{F}$, the stable kernel $\mathcal{F}^{sk}$ does not contain redundant, strictly strongly unacceptable, and strongly defeated arguments.*

We show that the stable kernels of two strongly equivalent cvAFs contain the same claims.

Lemma C.14. *For two cvAFs $\mathcal{F}$ and $\mathcal{G}$, $\mathcal{F} \equiv_s^{stb} \mathcal{G}$ implies $cl(A_{\mathcal{F}^{sk}}) = cl(A_{\mathcal{G}^{sk}})$.*

Proof. Let $x \in A_{\mathcal{F}^{sk}}$ with $cl(x) = c$. Towards a contradiction, assume that there is no argument $y \in A_{\mathcal{G}^{sk}}$ with $cl(y) = c$. Since we may assume $stb_{cl}(\mathcal{F}^{sk}) = stb_{cl}(\mathcal{G}^{sk})$ in this case we deduce that x does not occur in any stable extension of $\mathcal{F}^{sk}$.

Case 1 Suppose x is no self-attacker. We have to deal with three kinds of arguments:

- same claim as c (we block these arguments),
- attacking x (we block these arguments, whenever x cannot do this on its own),
- odd cycles (we disrupt all of them).

Then c appears in a stable extension of $\mathcal{F}$ but not in $\mathcal{G}$, because we will never add claim c.

Consider the set $\mathcal{Y} = \{y \in A_{\mathcal{F}^{sk}} \mid cl(y) = c, y \neq x\}$ of arguments with claim c. By assumption, none of these arguments attacks x. We consider arguments which defeat them unless this would require either defeating x as well or adding claim c. We let

$$V_{\mathcal{Y}} = \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}, e \neq c\}.$$

The remaining arguments $y \in \mathcal{Y}$ which are not attacked by $V_{\mathcal{Y}}$ must satisfy $vul(y) \setminus vul(x) = \{c\}$ and are therefore attacked by x (self-attackers).

Now consider the set $\mathcal{Z} = \{z \in A_{\mathcal{F}^{sk}} \mid (z, x) \in R_{\mathcal{F}}, (x, z) \notin R_{\mathcal{F}}\}$ of arguments attacking x without receiving a counter-attack. For $z \in \mathcal{Z}$ it holds that $cl(z) \in vul(x)$ and therefore, by our definition of the stable kernel, it cannot be the case that $vul(z) \subseteq vul(x)$. Moreover, $c \notin vul(z)$ since that would imply a counterattack from x. Therefore with

$$V_{\mathcal{Z}} = \{v_e = (\emptyset, e) \mid e \in vul(z) \setminus vul(x), z \in \mathcal{Z}\}$$

we get rid of them and we have now already ensured that $\{x\}$ becomes admissible. Observe that if $\mathcal{Z}$ is empty, then $V_{\mathcal{Z}}$ is empty as well.

Now consider the set $\mathcal{S} = \{s \in A_{\mathcal{F}} \mid (s, s) \in R_{\mathcal{F}}\}$ of self-attacking arguments. By definition of the stable kernel, we have $vul(s) \not\subseteq vul(x)$ for all $s \in \mathcal{S}$. Therefore with $V_{\mathcal{S}} = \{v_e \mid e \in vul(s) \setminus vul(x), s \in \mathcal{S}\}$ we get rid of them without attacking x.

Now consider any odd cycle $\mathcal{O} = \{o_1, \ldots, o_n\}$ occurring in $\mathcal{F}^{sk}$. Our goal is to argue that $\bigcup vul(o_i) \subseteq vul(x)$ is impossible; i.e., we can disrupt the odd cycle without attacking x. Assume the contrary, i.e., suppose $\bigcup vul(o_i) \subseteq vul(x)$. Then $cl(o_i) \in vul(x)$ for each i. Since $vul(o_i) \subseteq vul(x)$ this implies that x is unacceptable contradicting the construction of the stable kernel $\mathcal{F}^{sk}$. Thus, by adding appropriate arguments we can disrupt the odd cycles and therefore, the admissible set $\{x\}$ can be extended to a stable extension.

Case 2 Now suppose each argument with claim c is a self-attacker and fix such x. Again, we have to deal with three kinds of arguments:

- same claim as c (block these arguments),
- attacking x (we block all of these arguments),
- odd cycles (we disrupt all of them).

Then $\mathcal{F}$ has no stable extension, but one after we add claim c, where in $\mathcal{G}$ adding c does not change anything.

Consider the set $\mathcal{Y} = \{y \in A_{\mathcal{F}^{sk}} \mid cl(y) = c, y \neq x\}$ of arguments with claim c. We consider arguments which defeat them; this time we can get rid of all of them via

$$\begin{aligned} V_{\mathcal{Y}} &= \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}, e \neq c\} \\ &= \{v_e = (\emptyset, e) \mid e \in vul(y) \setminus vul(x), y \in \mathcal{Y}\}. \end{aligned}$$

Now consider the set $\mathcal{Z} = \{z \in A_{\mathcal{F}^{sk}} \mid (z, x) \in R_{\mathcal{F}^{sk}}\} \setminus \mathcal{Y}$ of arguments attacking x and not having claim c. For $z \in \mathcal{Z}$ it holds that $cl(z) \in vul(x)$ and therefore, by our definition of the stable kernel, it cannot be the case that $vul(z) \subseteq vul(x)$. Moreover, $c \in vul(x)$ implies $c \notin vul(z) \setminus vul(x)$. Therefore with

$$V_{\mathcal{Z}} = \{v_e = (\emptyset, e) \mid e \in vul(z) \setminus vul(x), z \in \mathcal{Z}\}$$

we get rid of them without introducing claim c. Moreover, we deal with the set of self-attackers $\mathcal{S} = \{s \in A_{\mathcal{F}} \mid (s, s) \in R_{\mathcal{F}}\}$ as before via $V_{\mathcal{S}} = \{v_e \mid e \in vul(s) \setminus vul(x), s \in \mathcal{S}\}$.

Now consider any odd cycle $\mathcal{O} = \{o_1, \ldots, o_n\}$ occurring in $\mathcal{F}^{sk}$. Again, our goal is to argue that $\bigcup vul(o_i) \subseteq vul(x)$ is impossible. Towards a contradiction, suppose $\bigcup vul(o_i) \subseteq vul(x)$. Then $cl(o_i) \in vul(x)$ for each i. Since $vul(o_i) \subseteq vul(x)$ this would, however, imply that x is unacceptable contradicting the construction of the stable kernel $\mathcal{F}^{sk}$. Thus, by adding appropriate arguments we ensure that $\mathcal{F}$ has no stable extension, but with the self-attacker x being the only odd cycle.

Therefore, $f_e(\mathcal{F}^{sk}, X)$ has no stable extension, but adding an isolated argument with claim c resolves this; meanwhile, adding this argument to $f_e(\mathcal{G}^{sk}, X)$ does not influence whether or not there is a stable extension. □

Appendix D. Logic Programs

Proposition 8.12. *Consider a semantics σ. Deciding atom-enforceability w.r.t. σ for the class of normal LPs is* NP*-hard.*

Proof. Let φ be a boolean formula given by clauses C over variables in X. The corresponding logic program P contains the following rules:

- the atomic rule '$p_\varphi \leftarrow \texttt{not}\ C$';
- rules '$p_c \leftarrow \{l \mid \neg l \in c\}, \texttt{not}\ \{l \in X \mid l \in c\}$' for each clause $c \in C$.

Intuitively, a clause-atom c is contained in a stable model M iff c is false in M. Hence we can accept φ iff $c \notin M$ for all $c \in C$. We show φ is satisfiable iff p_φ is enforceable in P. Since each stable model is well-founded in P it suffices to focus on stable semantics.

First assume φ is satisfiable. Assume M is a model of φ. We add each $x \in M$ as fact. We show that $Q = M \cup \{p_\varphi\}$ is a stable model of $P \cup M$. Consider $c \in C$. If $c \cap M \neq \emptyset$ then the rule r with $head(r) = p_c$ contains $\texttt{not}\ x$ for some $x \in c \cap M$. Hence the rule r is satisfied by Q. Likewise, if $c \cap M = \emptyset$ we have some $x \in X$ with $x \notin M$ and $\neg x \in c$. Hence the rule r with $head(r) = p_c$ satisfies $x \in body(r)$. Hence Q satisfies r.

For the other direction, assume p_φ is enforceable. Let R denote the set of rules which enforce p_φ, and let M denote the model of $R \cup P$ which contains p_φ. Then M does not contain any $c \in C$ (otherwise, p_φ would not be acceptable). Now, we show that $N = M \cap X$ is a model of φ. Again, for each rule $r \in P$ correspondig to a clause in $c \in C$, there is either some $x \in N$ with $\texttt{not}\ x \in body(r)$—in this case, $x \in c$ hence c is satisfied; or there is some $x \in X \setminus N$ with $x \in body(r)$—then $\neg x \in c$ and thus c is satisfied. □

References

Alferes, J. J., Leite, J. A., Pereira, L. M., Przymusinska, H., & Przymusinski, T. C. (2000). Dynamic updates of non-monotonic knowledge bases. *The journal of logic programming*, *45*(1-3), 43–70.

Amgoud, L., Besnard, P., & Vesic, S. (2014). Equivalence in logic-based argumentation. *Journal of Applied Non-Classical Logics*, *24*(3), 181–208.

Arora, S., & Barak, B. (2009). *Computational Complexity - A Modern Approach*. Cambridge University Press.

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

Baroni, P., Cerutti, F., Giacomin, M., & Guida, G. (2011). AFRA: Argumentation framework with recursive attacks. *International Journal of Approximate Reasoning*, *52*(1), 19–37.

Baroni, P., Gabbay, D. M., Giacomin, M., & van der Torre, L. (Eds.). (2018). *Handbook of Formal Argumentation*. College Publications.

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 *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, pp. 127–132.

Baumann, R., & Brewka, G. (2010). Expanding argumentation frameworks: Enforcing and monotonicity results. In *Proceedings of the 3rd Conference on Computational Models of Argument (COMMA 2010)*, Vol. 216 of *Frontiers in Artificial Intelligence and Applications*, pp. 75–86. IOS Press.

Baumann, R., Rapberger, A., & Ulbricht, M. (2022). Equivalence in argumentation frameworks with a claim-centric view - Classical results with novel ingredients. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, pp. 5479–5486. AAAI Press.

Baumann, R., & Strass, H. (2013). On the maximal and average numbers of stable extensions. In *Theory and Applications of Formal Argumentation - Second International Workshop, TAFA 2013, Beijing, China, August 3-5, 2013, Revised Selected papers*, Vol. 8306 of *Lecture Notes in Computer Science*, pp. 111–126. Springer.

Baumann, R., & Strass, H. (2022). An abstract, logical approach to characterizing strong equivalence in non-monotonic knowledge representation formalisms. *Artificial Intelligence*, *305*, 103680.

Baumann, R., & Ulbricht, M. (2021). On cycles, attackers and supporters - A contribution to the investigation of dynamics in abstract argumentation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI 2021)*, pp. 1780–1786. ijcai.org.

Baumeister, D., Järvisalo, M., Neugebauer, D., Niskanen, A., & Rothe, J. (2021). Acceptance in incomplete argumentation frameworks. *Artificial Intelligence*, *295*, 103470.

Baumeister, D., Neugebauer, D., Rothe, J., & Schadrack, H. (2018). Verification in incomplete argumentation frameworks. *Artificial Intelligence*, *264*, 1–26.

Bench-Capon, T. J. M., & Dunne, P. E. (2007). Argumentation in artificial intelligence. *Artificial Intelligence*, *171*(10-15), 619–641.

Besnard, P., & Hunter, A. (2001). A logic-based theory of deductive arguments. *Artificial Intelligence*, *128*(1-2), 203–235.

Besnard, P., & Hunter, A. (2018). A review of argumentation based on deductive arguments. In *Handbook of Formal Argumentation*, chap. 9, pp. 437–484. College Publications.

Borg, A., & Bex, F. (2021). Enforcing sets of formulas in structured argumentation. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 130–140.

Brass, S., & Dix, J. (1997). Characterizations of the disjunctive stable semantics by partial evaluation. *The Journal of Logic Programming*, *32*(3), 207–228.

Cabalar, P. (2002). A three-valued characterization for strong equivalence of logic programs. In *Proceedings of the 18th National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 2002)*, pp. 106–111. AAAI Press / The MIT Press.

Caminada, M., Sá, S., Alcântara, J., & 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., & Dvořák, W. (2015b). On the equivalence between logic programming semantics and argumentation semantics. *International Journal of Approximate Reasoning*, *58*, 87–111.

Cayrol, C., de Saint-Cyr, F. D., & Lagasquie-Schiex, M. (2010). Change in abstract argumentation frameworks: Adding an argument. *Journal of Artificial Intelligence Research*, *38*, 49–84.

Cayrol, C., & Lagasquie-Schiex, M.-C. (2005). On the acceptability of arguments in bipolar argumentation frameworks. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, Vol. 3571 of *Lecture Notes in Computer Science*, pp. 378–389. Springer.

Cyras, K., Fan, X., Schulz, C., & Toni, F. (2018). Assumption-based argumentation: Disputes, explanations, preferences. In *Handbook of Formal Argumentation*, chap. 7, pp. 365–408. College Publications. Also appears in IfCoLog Journal of Logics and their Applications 4(8):2407–2456.

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., & Woltran, S. (2015). Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence*, *228*, 153–178.

Dvorák, W., Fandinno, J., & Woltran, S. (2019). On the expressive power of collective attacks. *Argument and Computation*, *10*(2), 191–230.

Dvořák, W., & Dunne, P. E. (2018). Computational problems in formal argumentation and their complexity. In *Handbook of Formal Argumentation*, chap. 14, pp. 631–687. College Publications. Also appears in IfCoLog Journal of Logics and their Applications 4(8):2557–2622.

Dvořák, W., Greßler, A., Rapberger, A., & Woltran, S. (2023). The complexity landscape of claim-augmented argumentation frameworks. *Artificial Intelligence*, *317*, 103873.

Dvořák, W., Rapberger, A., & Woltran, S. (2019). Strong equivalence for argumentation frameworks with collective attacks. In *Proceedings of the 42nd German Conference on AI (KI 2019)*, Vol. 11793 of *Lecture Notes in Computer Science*, pp. 131–145. Springer.

Dvořák, W., Rapberger, A., & Woltran, S. (2020). Argumentation semantics under a claim-centric view: Properties, expressiveness and relation to SETAFs. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)*, pp. 341–350.

Dvořák, W., & Woltran, S. (2020). Complexity of abstract argumentation under a claim-centric view. *Artificial Intelligence*, *285*, 103290.

Dvořák, W., Rapberger, A., & Woltran, S. (2020). On the different types of collective attacks in abstract argumentation: equivalence results for SETAFs. *Journal of Logic and Computation*, *30*(5), 1063–1107.

Eiter, T., Fink, M., Tompits, H., & Woltran, S. (2004). Simplifying logic programs under uniform and strong equivalence. In *Proceedings of the 7th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2004)*, Vol. 2923 of *Lecture Notes in Computer Science*, pp. 87–99. Springer.

Eiter, T., Fink, M., Tompits, H., & Woltran, S. (2007). Complexity results for checking equivalence of stratified logic programs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp. 330–335.

Falappa, M. A., Kern-Isberner, G., & Simari, G. R. (2009). Belief revision and argumentation theory. In *Argumentation in Artificial Intelligence*, pp. 341–360. Springer.

Gabbay, D., Giacomin, M., Simari, G. R., & Thimm, M. (Eds.). (2021). *Handbook of Formal Argumentation*, Vol. 2. College Publications.

García, A. J., & Simari, G. R. (2018). Argumentation based on logic programming. In *Handbook of Formal Argumentation*, chap. 8, pp. 409–435. College Publications.

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

Hadjisoteriou, E., & Kakas, A. C. (2015). Reasoning about actions and change in argumentation. *Argument and Computation*, *6*(3), 265–291.

Janhunen, T. (2004). Representing normal programs with clauses. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI 2004)*, pp. 358–362. IOS Press.

Kaci, S., van der Torre, L. W. N., Vesic, S., & Villata, S. (2021). Preference in abstract argumentation. In *Handbook of Formal Argumentation, Volume 2*, pp. 211–248. College Publications.

König, M., Rapberger, A., & Ulbricht, M. (2022). Just a matter of perspective: Intertranslating expressive argumentation formalisms. In *Proceedings of the 10th International Conference on Computational Models of Argument (COMMA 2022)*, Vol. 353 of *Frontiers in Artificial Intelligence and Applications*, pp. 212–223. IOS Press.

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

Lin, F. (2002). Reducing strong equivalence of logic programs to entailment in classical propositional logic. In *Proceedings of the 8th International Conference on Principles and Knowledge Representation and Reasoning (KR 2002)*, pp. 170–176. Morgan Kaufmann.

Lin, F., & Chen, Y. (2007). Discovering classes of strongly equivalent logic programs. *Journal of Artificial Intelligence Research*, *28*, 431–451.

Modgil, S., & Prakken, H. (2018). Abstract rule-based argumentation. In *Handbook of Formal Argumentation*, chap. 6, pp. 287–364. College Publications. Also appears in IfCoLog Journal of Logics and their Applications 4(8):2319–2406.

Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., García, A. J., & Simari, G. R. (2008). Argument theory change applied to defeasible logic programming. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2008)*, pp. 132–137. AAAI Press.

Nielsen, S. H., & Parsons, S. (2006). A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *3rd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006), Revised Selected and Invited Papers*, Vol. 4766 of *Lecture Notes in Computer Science*, pp. 54–73. Springer.

Nielsen, S. H., & Parsons, S. (2007). An application of formal argumentation: Fusing bayesian networks in multi-agent systems. *Artificial Intelligence*, *171*(10-15), 754–775.

Odekerken, D., Bex, F., Borg, A., & Testerink, B. (2022). Approximating stability for applied argument-based inquiry. *Intelligent Systems with Applications*, *16*, 200110.

Oikarinen, E., & Woltran, S. (2011). Characterizing strong equivalence for argumentation frameworks. *Artificial Intelligence*, *175*(14-15), 1985–2009.

Osorio, M., Pérez, J. A. N., & Arrazola, J. (2001). Equivalence in answer set programming. In *Proceedings of the 11th International Workshop on Logic Based Program Synthesis and Transformation (LOPSTR 2001)*, Vol. 2372 of *Lecture Notes in Computer Science*, pp. 57–75. Springer.

Papadimitriou, C. H. (1994). *Computational complexity*. Addison-Wesley.

Pearce, D., Tompits, H., & Woltran, S. (2001). Encodings for equilibrium logic and logic programs with nested expressions. In *Progress in Artificial Intelligence, Knowledge Extraction, Multi-agent Systems, Logic Programming and Constraint Solving, Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA 2001)*, Vol. 2258 of *Lecture Notes in Computer Science*, pp. 306–320. Springer.

Prakken, H. (2022). Formalising an aspect of argument strength: Degrees of attackability. In Toni, F., Polberg, S., Booth, R., Caminada, M., & Kido, H. (Eds.), *Proceedings of the 10th International Conference on Computational Models of Argument (COMMA 2022)*, Vol. 353 of *Frontiers in Artificial Intelligence and Applications*, pp. 296–307. IOS Press.

Rapberger, A. (2020). Defining argumentation semantics under a claim-centric view. In *Proceedings of the 9th European Starting AI Researchers' Symposium (STAIRS 2020)*, Vol. 2655 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rapberger, A., & Ulbricht, M. (2022). On dynamics in structured argumentation formalisms. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR 2022)*, pp. 288–298.

Rapberger, A., Ulbricht, M., & Wallner, J. P. (2022). Argumentation frameworks induced by assumption-based argumentation: Relating size and complexity. In *Proceedings* *of the 20th International Workshop on Non-Monotonic Reasoning (NMR 2022)*, Vol. 3197 of *CEUR Workshop Proceedings*, pp. 92–103. CEUR-WS.org.

Rotstein, N. D., Moguillansky, M. O., Falappa, M. A., García, A. J., & Simari, G. R. (2008). Argument theory change: Revision upon warrant. In *Proceedings of the 2nd International Conference on Computational Models of Argument: Proceedings of (COMMA 2008)*, Vol. 172 of *Frontiers in Artificial Intelligence and Applications*, pp. 336–347. IOS Press.

Rotstein, N. D., Moguillansky, M. O., García, A. J., & Simari, G. R. (2010). A dynamic argumentation framework. In *Proceedings of the 3rd International Conference on Computational Model of Argument (COMMA 2010)*, Vol. 216 of *Frontiers in Artificial Intelligence and Applications*, pp. 427–438. IOS Press.

Sá, S., & Alcântara, J. F. L. (2021). Assumption-based argumentation is logic programming with projection. In *Proceedings of the 16th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2021)*, Vol. 12897 of *Lecture Notes in Computer Science*, pp. 173–186. Springer.

Snaith, M., & Reed, C. (2017). Argument revision. *Journal of Logic and Computation*, *27*(7), 2089–2134.

Testerink, B., Odekerken, D., & Bex, F. (2019). A method for efficient argument-based inquiry. In *Proceedings of the 13th International Conference on Flexible Query Answering Systems (FQAS 2019)*, pp. 114–125. Springer.

Thimm, M. (2012). A probabilistic semantics for abstract argumentation. In *Proceeedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, Vol. 242 of *Frontiers in Artificial Intelligence and Applications*, pp. 750–755. IOS Press.

Turner, H. (2001). Strong equivalence for logic programs and default theories (made easy). In *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2001)*, Vol. 2173 of *Lecture Notes in Computer Science*, pp. 81–92. Springer.

Ulbricht, M. (2021). On the maximal number of complete extensions in abstract argumentation frameworks. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021)*, pp. 707–711.

Ulbricht, M., & Baumann, R. (2019). If nothing is accepted - Repairing argumentation frameworks. *Journal of Artificial Intelligence Research*, *66*, 1099–1145.

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

Wallner, J. P., Niskanen, A., & Järvisalo, M. (2017). Complexity results and algorithms for extension enforcement in abstract argumentation. *Journal of Artificial Intelligence Research*, *60*, 1–40.

Wang, K., & Zhou, L. (2005). Comparisons and computation of well-founded semantics for disjunctive logic programs. *ACM Transactions on Computational Logic*, *6*(2), 295–327.