Enforcement in Formal Argumentation

Ringo Baumann
Department of Computer Science, Leipzig University, Germany
baumann@informatik.uni-leipzig.de

Sylvie Doutre
Institut de Recherche en Informatique de Toulouse, Université Toulouse Capitole, France
doutre@irit.fr

Jean-Guy Mailly
Laboratoire d'Informatique Paris Descartes, Université de Paris, France
jean-guy.mailly@u-paris.fr

Johannes P. Wallner
Institute of Software Technology, Graz University of Technology, Austria
wallner@ist.tugraz.at

Abstract

Within argumentation dynamics, a major strand of research is concerned with how changing an argumentation framework affects the acceptability of arguments, and how to modify an argumentation framework in order to guarantee that some arguments have a given acceptance status. In this chapter, we overview the main approaches for enforcement in formal argumentation. We mainly focus on extension enforcement, i.e., on how to modify an argumentation framework to ensure that a given set of arguments becomes (part of) an extension. We present different forms of extension enforcement defined in the literature, as well as several possibility and impossibility results. The question of minimal change is also considered, i.e., what is the minimal number of modifications that must be made to the argumentation framework for enforcing an extension. Computational complexity and algorithms based on a declarative approach are discussed. Finally, we briefly describe several notions that do not directly fit our definition of extension enforcement, but are closely related.

1 Introduction

At the beginning of the 2010s several problems regarding *dynamic aspects* of abstract argumentation have been addressed in the literature [30; 38; 28; 65]. One much cited problem among these is the so-called *enforcing problem* dealing with changing the acceptability of certain arguments [17]. Over the years, the problem gained more and more attention which finally leads to the writing of this chapter. In its very first version the problem can be briefly summarized as the question whether it is possible, given a specific type of syntactic changes, to modify a given AF such that a desired set of arguments becomes (a subset of) an extension. Consider the following snapshot of a dialogue among agents A and B depicted in Figure 1. Assume it is A's turn and her desired set of arguments is $E = \{a_1, a_2, a_3\}$. Furthermore, A and B are discussing under preferred semantics, which selects maximal conflict-free and self-defending sets of arguments.

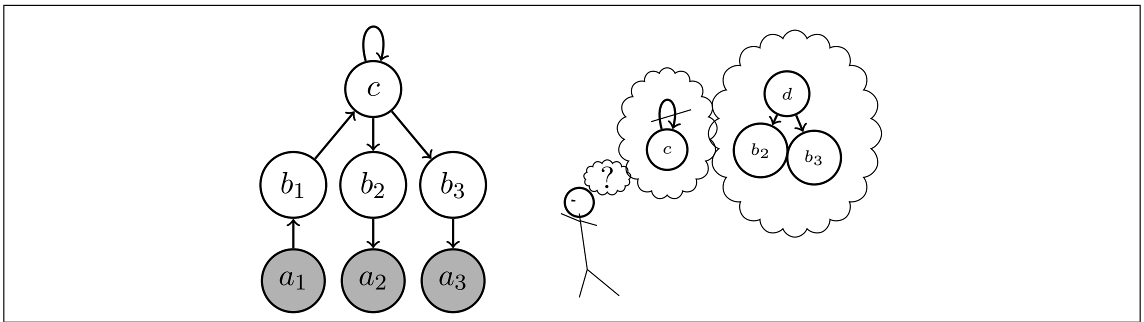


Figure 1: Snapshot of a dialogue

In order to enforce E, agent A may come up with new arguments which interact with the old ones (for example through introducing an argument d which attacks b_2 and b_3) and/or question old arguments or attacks between them, respectively (for example through questioning the self-attack of c). Please note that first, in this scenario, enforcing is possible and second, that there are at least two different possibilities to achieve that. This insight leads to a further well-studied issue, namely the so-called *minimal change problem* firstly introduced in [13]. This problem is defined as a generalization of the classical enforcing problem since one is not only interested in whether enforcements are *possible*, but also in the *effort needed* to enforce a set of arguments. One numerical measure which is frequently used for this effort corresponds to the number of additions or removals of attacks to reach such an enforcement. The main motivation behind this measure is that adding or removing an isolated argument does not contribute at all to solving or increasing a given conflict, i.e. the conflicting information remains the same. This means, the decrease or increase of a conflict is directly linked to upcoming or disappearing attacks and thus, counting attacks only is a reasonable approach. Regarding the introductory example we obtain a minimal effort of 1 if allowing arbitrary modifications.

In this chapter we give an overview over main variants of enforcement studied in the literature. We give a particular focus on strict and non-strict extension enforcement, whose aim is to modify an AF such that a desired set of arguments becomes exactly (or part of) an extension, under a semantics. A main distinguishing factor among the family of operators for extension enforcement is how an AF may be modified. We highlight here changes corresponding to expansions, i.e., additions of arguments and attacks such as the addition of argument d above, or local updates, i.e., modifying only the attack structure such as questioning the self-attack of c, but also discuss modifications to AFs more broadly, as well. Additionally, we consider as an instance of a change that does not affect the structure of the framework, modifications of the chosen semantics, in order to enforce a set of arguments.

We present main formal properties of extension enforcement derived in the literature, e.g., for impossibility and possibility results, and results for the minimal change problem of extension enforcement. We further survey results regarding the complexity of reasoning on enforcement and present algorithms based on declarative approaches to implement enforcement.

The chapter starts off with recalling formal preliminaries of AFs (Section 2) including types of modifications on AFs. The main section on extension enforcement is Section 3, which first introduces enforcement as a general problem, and focuses on the extension enforcement variant. In this section, we present expansion-based extension enforcement and extension enforcement based on locally updating an attack structure without modifying the set of arguments. Further, minimal change, semantics change, complexity results, and algorithms, are presented. In Section 4 we survey related notions to enforcement, and we close with a discussion of related works (Section 4.5) and with conclusions (Section 5).

2 Formal Preliminaries

In order to keep the chapter self-contained we review all relevant definitions. We start with the basic notions of Dung's abstract argumentation theory [54].

2.1 Argumentation Frameworks and Semantics

An *abstract argumentation framework (AF)* is just a directed graph $F = (A, R)$ where a node $a \in A$ is called an *argument* and a pair $(a, b) \in R \subseteq A \times A$ is interpreted as an *attack* from argument a to argument b. We require that any AF

$F = (A, R)$ possesses arguments from a fixed reference set $\mathcal{U}$, i.e. $A \subseteq \mathcal{U}$. Moreover, in this chapter we restrict ourselves to finite AFs, i.e. any AF consists of finitely many arguments and attacks only. Note that this is a common restriction in the literature although actual and potential infinite AFs play an important role in practical applications as well as theoretical considerations (cf. [8; 22; 16] for more information). At the heart of Dung's abstract argumentation theory are *argumentation semantics* which formalize intuition of what should be acceptable in the light of conflicts. Two main approaches to argumentation semantics can be found, namely so-called *extension-based* and *labelling-based* versions (cf. [7] for an introduction and [10, Sections 2.2, 4.4] for further relations). In this chapter we concentrate on the former only. Consider the following generic definition. The set $\mathcal{F}$ refers to all considered AFs.

Definition 2.1. *A semantics is a total function*

$$\sigma : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto \sigma(F) \subseteq 2^A.$$

A set of arguments $E \in \sigma(F)$ is called a *σ-extension*. Moreover, we say that a semantics σ is *universally defined* if each AF admits at least one extension with respect to this semantics, i.e. for any $F \in \mathcal{F}$, $|\sigma(F)| \geq 1$. Furthermore, a semantics σ is said to be *uniquely defined* if always exactly one set of arguments is returned, i.e. $|\sigma(F)| = 1$ for any $F \in \mathcal{F}$.

Before presenting the relevant semantics for this chapter we have to introduce some further notation. Given an AF $F = (A, R)$ and a set $E \subseteq A$. We use E^+_F, or simply E^+, for $\{b \mid (a, b) \in R, a \in E\}$. Moreover, $E^\oplus_F$, or simply $E^\oplus$, is called the *range* of E and stands for $E^+ \cup E$. Analogously, E^-_F (or simply E^-) stands for $\{b \mid (b, a) \in R, a \in E\}$, and $E^\ominus_F$ (or simply $E^\ominus$) corresponds to $E^- \cup E$. An argument a is *defended* by E (in F) if for each $b \in A$ with $(b, a) \in R$, b is attacked by some $c \in E$. Finally, $\Gamma_F : 2^A \to 2^A$ with $I \mapsto \{a \in A \mid a \text{ is defended by } I\}$ denotes the so-called *characteristic function* (of F).

Besides conflict-free and admissible sets (abbreviated by *cf* and *ad*) we consider a large number of well-known semantics, namely naive, stage, stable, semi-stable, complete, preferred, grounded, ideal, and eager semantics (abbreviated by *na*, *stg*, *stb*, *sst*, *co*, *pr*, *gr*, *id*, *eg*, respectively).

Definition 2.2. *Let $F = (A, R)$ be an AF and $E \subseteq A$.*

1. *$E \in cf(F)$ iff for no $a, b \in E$, $(a, b) \in R$,*
2. *$E \in na(F)$ iff E is $\subseteq$-maximal in $cf(F)$,*
3. *$E \in stg(F)$ iff $E \in cf(F)$ and $E^\oplus$ is $\subseteq$-maximal in $\{I^\oplus \mid I \in cf(F)\}$,*

4. *$E \in stb(F)$ iff $E \in cf(F)$ and $E^{\oplus} = A$,*

5. *$E \in ad(F)$ iff $E \in cf(F)$ and $E \subseteq \Gamma_F(E)$,*

6. *$E \in sst(F)$ iff $E \in ad(F)$ and $E^{\oplus}$ is $\subseteq$-maximal in $\{I^{\oplus} \mid I \in ad(F)\}$,*

7. *$E \in co(F)$ iff $E \in cf(F)$ and $E = \Gamma_F(E)$,*

8. *$E \in pr(F)$ iff E is $\subseteq$-maximal in $co(F)$,*

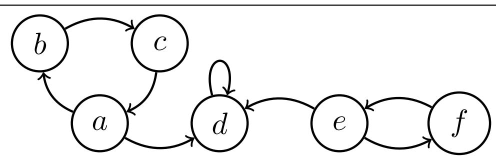
9. *$E \in gr(F)$ iff E is $\subseteq$-minimal in $co(F)$,*

10. *$E \in id(F)$ iff E is $\subseteq$-maximal in $\{I \mid I \in ad(F), I \subseteq \bigcap pr(F)\}$,*

11. *$E \in eg(F)$ iff E is $\subseteq$-maximal in $\{I \mid I \in ad(F), I \subseteq \bigcap sst(F)\}$.*

It has been shown that any of the introduced semantics is universally defined except the stable one and moreover, grounded, ideal and eager semantics are even uniquely defined (cf. [21] for an overview). In order to get familiar with the introduced definitions consider the following example taken from [15].

Example 2.3. *Consider the AF $F = (A, R)$ with $A = \{a, b, c, d, e, f\}$ and $R = \{(a, b), (a, d), (b, c), (c, a), (d, d), (e, d), (e, f), (f, e)\}$. The graphical representation of F is given below.*



The evaluation of F w.r.t. the introduced semantics is given in the following table. The entry "✓" in row "σ" and line "E" stands for $E \in \sigma(F)$.

	stb	sst	stg	pr	ad	co	gr	id	eg	na
$\emptyset$					✓	✓	✓	✓		
$\{e\}$		✓		✓	✓	✓			✓	
$\{f\}$				✓	✓	✓				
$\{a,e\}$			✓							✓
$\{b,e\}$			✓							✓
$\{c,e\}$			✓							✓
$\{a,f\}$			✓							✓
$\{b,f\}$										✓
$\{c,f\}$										✓

Table 1: Evaluation table of F

The AF F is an example for a collapse of stable semantics, i.e. $stb(F) = \emptyset$. The non-existence of stable extensions in F implies the occurrence of odd-length cycles like the 3-cycle $[a, b, c, a]$ or the self-loop $[d, d]$. More precisely, in case of finite AFs we have that being odd-cycle free is sufficient for warranting at least one stable extension [54; 81].

As already indicated in Table 1 there are several well-known subset relations between the considered semantics. For instance, for any AF F we have, $stb(F) \subseteq sst(F) \subseteq pr(F) \subseteq co(F) \subseteq ad(F)$ and $stb(F) \subseteq stg(F) \subseteq na(F)$.

2.2 Acceptance Modes and Structural Changes

In the following we present several acceptance modes and structural changes, that is, changes on the structure (addition or removal of arguments and attacks) of the AF, which can be used to specify a certain type of enforcement.

So-called *credulous* and *sceptical acceptance* are the most common reasoning types for abstract argumentation semantics. They are usually defined for single arguments only. We present their definitions for sets of arguments where the classical single argument acceptance can be obtained by considering the singleton of the argument in question. Moreover, since a non-universally defined semantics σ may return no σ-extension for a given AF F we consider so-called *non-empty sceptical reasoning* which avoids the (possibly) unintended situation that every argument is sceptically accepted due to the emptiness of $\sigma(F)$. A further frequently used acceptance mode is the requirement to be contained in at least one extension, so-called *covered acceptance*[1]. This notion plays a central role in the field of enforcement and is located in-between non-empty sceptical and credulous acceptance.

Definition 2.4. *Given a semantics σ, an AF $F = (A, R)$ and a set $E \subseteq A$. We say that E is*

1. *credulously accepted w.r.t. σ if $E \subseteq \bigcup \sigma(F)$,*
2. *sceptically accepted w.r.t. σ if $E \subseteq \bigcap \sigma(F)$,*
3. *non-empty sceptically accepted w.r.t. σ if $E \subseteq \bigcap \sigma(F)$ and $\sigma(F) \neq \emptyset$,*
4. *covered accepted w.r.t. σ if there is an $E' \in \sigma(F)$, s.t. $E \subseteq E'$.*

For convenience we introduce the following unified notation. We write $E \in \mathit{cred}(F, \sigma)$, $E \in \mathit{scep}(F, \sigma)$, $E \in \mathit{scep}^{\neq\emptyset}(F, \sigma)$ or $E \in \mathit{cov}(F, \sigma)$ for E is credulously, sceptically, non-empty sceptically or covered accepted, respectively. Moreover, for any given reasoning type r we use $E \in r_s(\sigma, F)$ to indicate that there is an equality instead of a subset relation only, e.g. there is an $E' \in \sigma(F)$, s.t. $E = E'$ in the case of covered acceptance (or, said otherwise, $E \in \sigma(F)$). In this case we say that the considered set E is *strictly* accepted. If E is non-empty sceptically accepted w.r.t. σ then E is covered accepted w.r.t. σ (since E must be part of all σ-extensions and there is at least one), and the latter implies that E is credulously accepted w.r.t. σ (since the witness for being covered accepted is a witness for credulous acceptance).

Let us proceed with the running AF exemplifying several acceptance modes.

Example 2.5 (Example 2.3 cont.)**.** *Let $\sigma = \mathit{stb}$. Since $\mathit{stb}(F) = \emptyset$ we obtain $\bigcup \mathit{stb}(F) = \emptyset$ and $\bigcap \mathit{stb}(F) = \mathcal{U}$. Hence, any set $E \subseteq \mathcal{U}$ is sceptically, but not non-empty sceptically accepted, i.e. $E \in \mathit{scep}(F, \mathit{stb})$ and $E \notin \mathit{scep}^{\neq\emptyset}(F, \mathit{stb})$. Moreover, E is neither credulously, nor covered accepted, i.e. $E \notin \mathit{cred}(F, \mathit{stb})$ and*

[1]We mention that this notion is sometimes called credulous acceptance [55, p. 704]. This is due to the fact that that there are at least two options if generalizing credulous acceptance from arguments to sets of arguments.

$E \notin cov(F, stb)$.
Consider now $\sigma = pr$. Since $pr(F) = \{\{e\}, \{f\}\}$ we have $\bigcup pr(F) = \{e, f\}$ and $\bigcap pr(F) = \emptyset$. Thus, $\{e, f\}$ is credulously strict but neither sceptically nor non-empty sceptically accepted, i.e. $\{e, f\} \in cred_s(F, pr)$, $\{e, f\} \notin scep(F, pr)$ and $\{e, f\} \notin scep^{\neq\emptyset}(F, pr)$. Moreover, $\{e, f\}$ is not covered accepted whereas $\{e\}$ and $\{f\}$ are and this acceptance is even strict, i.e. $\{e, f\} \notin cov(F, pr)$ and $\{e\}, \{f\} \in cov_s(F, pr)$.

We now introduce typical structural changes. The most general form of dynamic scenarios are so-called *updates* where arguments and attacks can be deleted and added. If we do not delete any information we call the structural change an *expansion* [17; 76; 12]. The following kinds of expansions have received particular attention in the literature. *Normal expansions* add new arguments and possibly new attacks which concern at least one of the fresh arguments. Moreover, *local expansions* do not introduce any new arguments but possibly new attacks among the old arguments. Both types of expansions naturally occur in the context of instantiation-based argumentation [27; 35]. For instance, adding a new piece of information to the underlying knowledge base corresponds to a normal expansion on the AF level. Furthermore, changing the considered notion of attack left the constructed arguments untouched and results in a local expansion. Two further subconcepts of normal expansions are usually considered, so-called *strong* and *weak expansions*. Their names refer to properties of the additional arguments, namely arguments which are never attacked by former arguments (*strong* arguments) and arguments which do not attack former arguments (*weak* arguments). The former type typically occurs in a debate if one tries to strengthen the own point of view via rebutting the opponents arguments. Note that weak expansions seem to be more an academic exercise than a task with practical relevance with regard to real-world argumentation. However, they do play a decisive role in the context of *splittings* [11; 19; 6].

Consider the formal definition of the discussed types of expansions.

Definition 2.6. *An AF G is an expansion of AF $F = (A, R)$ (for short, $F \preceq_E G$) iff $G = (A \cup B, R \cup S)$ for some (maybe empty) sets B and S, s.t. $A \cap B = R \cap S = \emptyset$. An expansion is called*

1. *normal ($F \preceq_N G$) iff $\forall ab \ \ ((a, b) \in S \rightarrow a \in B \vee b \in B)$,*
2. *strong ($F \preceq_S G$) iff $F \preceq_N G$ and $\forall ab \ \ ((a, b) \in S \rightarrow \neg(a \in A \wedge b \in B))$,*
3. *weak ($F \preceq_W G$) iff $F \preceq_N G$ and $\forall ab \ \ ((a, b) \in S \rightarrow \neg(a \in B \wedge b \in A))$,*
4. *local ($F \preceq_L G$) iff $B = \emptyset$.*

Example 2.7. *Consider the following simple AF F. The presented AFs F_X represent examples for $F \preceq_X F_X$. This means, F_N is a normal expansion of F. Note that grey-highlighted arguments or attacks represent added information.*

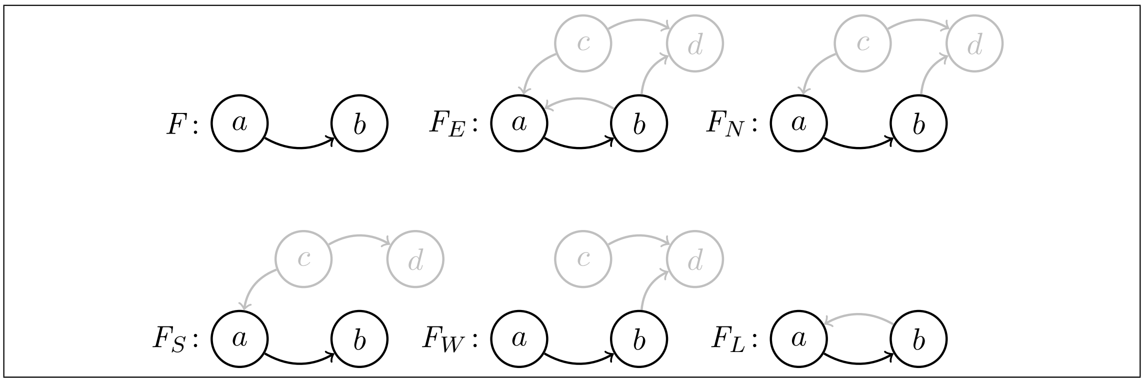


Figure 2: Different kinds of expansions

The natural counter-parts to expansions are so-called *deletions* where no further arguments and attacks are added [30; 28; 14]. We consider two sub-classes of *deletions* representing the inverse operations to normal and local expansions, namely *normal* and *local deletions*. Normal deletions retract arguments and their corresponding attacks. Such a kind of structural change occurs in the instantiation-based context if we delete information from the underlying knowledge base. Changing to a more restrictive notion of attack corresponds to a local deletion where only attacks are discarded.

In order to present the precise formal meaning of deletions we have to introduce some operations on directed graphs first. First, we use $F \sqcup H$ for the pointwise union of two AFs. In Definition 2.8, such an union is used in order to represent the addition of information (encoded in H) to an initial AF (F). Secondly, the restriction of $F = (A, R)$ to a set $B \subseteq A$ abbreviated as $F|_B$ is given via $(B, R \cap (B \times B))$.

Definition 2.8. *Given an AF $F = (A, R)$, a set of arguments B and a set of attacks S as well as a further AF H. The AF*

$$G = (F \setminus [B, S]) \sqcup H := \Big((A, R \setminus S)|_{A \setminus B}\Big) \sqcup H$$

is called an update of F (for short, $F \asymp_U G$). An update is called a

1. *deletion ($F \succeq_D G$) iff $H = (\emptyset, \emptyset)$,*
2. *normal deletion ($F \succeq_{ND} G$) iff $F \succeq_D G$ and $S = \emptyset$,*

3. *local deletion* ($F \succeq_{LD} G$) *iff* $F \succeq_D G$ *and* $B = \emptyset$.

Let us take a closer look at the definition of $G = (F \setminus [B, S]) \sqcup H$. The AF H plays the role of added information, i.e. it contains new arguments and attacks. Consequently, for all kind of deletions we have $H = (\emptyset, \emptyset)$ which leaves us with $G = F \setminus [B, S]$. The set B contains arguments which have to be deleted. Since attacks depend on arguments, we have to delete the attacks which involve arguments from B too. This operation is formally captured by the restriction of F to $A \setminus B$. Furthermore, the set S contains particular attacks which have to be deleted. This means, the pair $[B, S]$ does not necessarily have to be an AF. Therefore we use $[B, S]$ instead of (B, S). If clear from context we use B and S instead of $[B, \emptyset]$ or $[\emptyset, S]$, i.e. we simply write $F \setminus B$ as well as $F \setminus S$ for normal or local deletions, respectively. Note that the different kinds of expansion presented in Definition 2.6 can be captured by setting $B = S = \emptyset$. Deletions and expansions are dual concepts: $F \preceq_E G$ if and only iff $G \succeq_D F$, and similarly for the normal or local versions.

Example 2.9. *The AF F represents the initial situation. An update as well as arbitrary, normal or local deletion of it are given by F_U, F_D, F_{ND} and F_{LD}. Grey-highlighted arguments or attacks represent added information in contrast to dotted arguments and attacks which represent deleted objects.*[2] *More formally, in accordance with Definition 2.8 we have that $F_U = (F \setminus [B, S]) \sqcup H$, $F_D = F \setminus [B, S]$, $F_{ND} = F \setminus B$, $F_{LD} = F \setminus S$ where the set of arguments $B = \{c\}$, the set of attacks $S = \{(b, a)\}$ and the AF $H = (\{b, d, e, f\}, \{(d, b), (e, f), (f, d)\})$.*

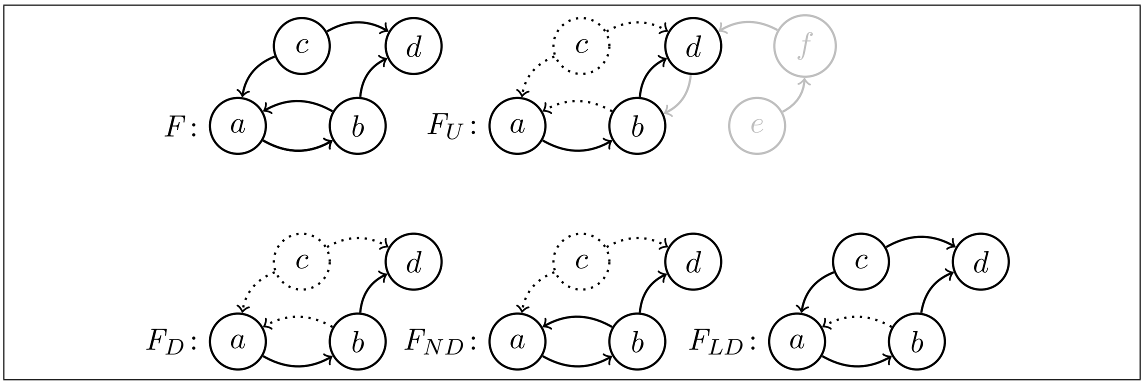


Figure 3: An update and different kinds of deletions

[2]This convention will be used throughout the whole chapter.

3 Enforcement

3.1 The General Setup

The starting point of any extension enforcement case is:

- an AF F,
- a semantics σ,
- a certain desired set of arguments E, together with
- a reasoning, acceptance mode r, e.g. credulous, sceptical, non-empty sceptical, covered, with a strict or non-strict goal achievement (cf. Section 2.2).

In addition, parameters indicating the way of achieving the enforcement can be specified, namely:

- allowed types of structural changes like update, expansion and deletion (cf. Section 2.2),
- allowed types of semantic changes, if any (cf. Section 3.2.4), and
- whether these changes would have to be minimal, and in which sense (cf. Section 3.2.3).

For illustrative purposes let us assume that r stands for credulous acceptance. Consequently, enforcement is needed if and only if E is not credulously accepted w.r.t. σ in F, i.e. $E \notin \mathit{cred}(F, \sigma)$. This is why we often speak of the *desired* set of arguments E since we want to fix the defect of non-acceptance. In order to achieve this goal we have two main options, namely structural changes and/or semantic changes. More precisely, we are looking for changes of AFs, from F to G, and/or semantics, from σ to τ, s.t. E is credulously accepted w.r.t. τ in G, i.e. $E \in \mathit{cred}(G, \tau)$. The way of how to perform the structural change is fixed in advance. For instance, one may require that only local expansions of F are allowed, i.e. $F \preceq_L G$. The same applies to the semantic change. One may allow changes to any kind of semantics or to admissibility-based ones only. Another option would be to completely forbid semantic changes, i.e. $\tau = \sigma$. In the following definition, we call a *modification type* $M \subseteq \mathcal{F} \times \mathcal{F}$ a relation such that $(F, G) \in M$ iff, when F is an initial AF, then G is a possible result of modifying F. For instance, $M = \preceq_L$ means that only local expansions are authorized.

Consider the following formal definition of an enforcement.

Definition 3.1. *Given two AFs F and G, two semantics σ and τ, a modification type $M \subseteq \mathcal{F} \times \mathcal{F}$, a set of argument E, and a reasoning mode r. A pair (G, τ) is called an (F, σ, M, r)-*enforcement *of E if*

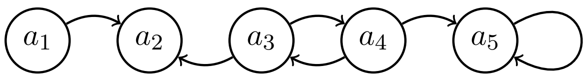
1. *$(F, G) \in M$ and*
2. *$E \in r(G, \tau)$.*

*Moreover, we call G the τ-*enforcing AF *and we say that E is τ-*enforced by G.

The different kinds of expansions and deletions presented in Definitions 2.6, 2.8 are captured by setting $M \in \{\preceq_E, \preceq_N, \preceq_S, \preceq_W, \preceq_L, \asymp_U, \succeq_D, \succeq_{ND}, \succeq_{LD}\}$. Note that $F \preceq_N G$ can be equivalently rewritten as $(F, G) \in \preceq_N$ since $\preceq_N$ is formally a binary relation over $\mathcal{F}$, i.e. $\preceq_N \subseteq \mathcal{F} \times \mathcal{F}$. Whenever F, σ, M and r are clear from context we simply speak of enforcements of E. If the set in question is strictly accepted we speak about a *strict* enforcement (for instance, $r = cov_s$), otherwise *non-strict* (for instance, $r = cov$). Moreover, we distinguish between *conservative* ($\sigma = \tau$) and *liberal* enforcements ($\sigma \neq \tau$). The latter may be interpreted as a change of proof standard or paradigm shift. Imagine a judicial proceeding. Here it is vitally important whether you are accused on the base of criminal or civil law. The required evidence is different and hence the acceptable sets of arguments differ.

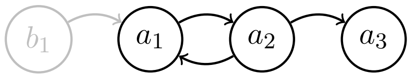
Consider the following two examples taken from [17].

Example 3.2 (liberal, strict)**.** *Given F as presented below, $\sigma = stb$, $M = \asymp_U$, $r = cov_s$ and the desired set $E = \{a_1, a_3\}$.*



Since $stb(F) = \{\{a_1, a_4\}\}$ we have $E \notin cov_s(stb, F)$. How to enforce E? Define an enforcement (G, τ) of E with $F = G$ and $\tau = pr$. Note that $pr(G) = \{\{a_1, a_3\}, \{a_1, a_4\}\}$ justifies the claim because $E \in cov_s(G, pr)$ holds. The considered enforcement is strict and liberal and F is the pr-enforcing AF.

Example 3.3 (conservative, non-strict)**.** *Given $\sigma = gr$, $M = \preceq_S$, $r = cov$, $E = \{a_2\}$ and $F = (\{a_1, a_2, a_3\}, \{(a_1, a_2), (a_2, a_1), (a_2, a_3)\})$ as presented below.*



Note that $gr(F) = \{\emptyset\}$. Hence, $E \notin cov(gr, F)$. In this example we allow strong expansions only. Is it possible to enforce E? The answer is "yes". Consider the enforcement (G, τ) of E with G defined as depicted above and $\tau = \sigma$. Since $gr(G) = \{\{b_1, a_2\}\}$ we deduce $E \in cov(G, gr)$. The considered enforcement is non-strict and conservative and G is the gr-enforcing AF.

3.2 Extension Enforcement with Structural Change

We start with a review of one of the most prominent enforcement operators in the literature, named extension enforcement [17; 13; 43; 52; 85; 60]. Extension enforcement refers to a family of enforcement operators that all deal with covered acceptance, i.e., the enforcement goal is to modify a given AF such that a desired set of arguments becomes an extension, or becomes part of an extension, under a semantics. Both strict and non-strict variants were studied.

The main distinguishing aspect of various extension enforcement operations is what kind of modification type is permitted. Concretely, we look at extension enforcement allowing only expansions (Section 3.2.1), permitting only local modifications (Section 3.2.2), i.e., changing the attack structure, restricting change to be minimal (Section 3.2.3), and changing semantics (Section 3.2.4).

3.2.1 Expansion-based enforcement

In this section we consider conservative (non-)strict enforcements w.r.t. covered reasoning mode under different forms of expansions. More precisely, for a given AF $F = (A, R)$, a semantics σ and a desired set of arguments $E \subseteq A$ we look at pairs (G, σ) being (F, σ, M, cov) enforcements of E. We allow $M \in \{\preceq_E, \preceq_N, \preceq_S, \preceq_W\}$, i.e. arbitrary, normal, strong, and weak expansions are considered. In the following, for the sake of brevity, we do not explicitly mention the covered acceptance mode as well as the conservativeness.

We have already seen a case of non-strict extension enforcement under strong expansions in Example 3.3. We now exemplify some properties of extension enforcement under expansions.

Example 3.4. *Let us consider an AF $F = (A, R)$ with $A = \{a, b, c, d\}$ and an attack relation as shown in Figure 4. Say we want to enforce $E = \{b, d\}$ to be part of an admissible extension in a non-strict manner, and allowing arbitrary expansions. An AF G that ad-enforces these constraints is shown, as well, in Figure 4. That is, expanding by two arguments e and f and adding attacks (b, f), (f, d), and (e, c) results in $\{e, b, d\} \in ad(G)$, and thus E is non-strictly enforced to be part of an admissible extension by G. Note that adding the single attack (d, c) only wouldn't* *do the job since we are interested in non-strict enforcements. However, there are many more ways to non-strictly enforce the desired set E. We encourage the reader to find other witnessing ad-enforcing expansions.*

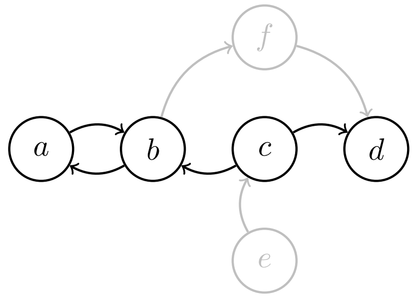


Figure 4: AF and expanded AF from Example 3.4

The preceding example illustrates the existence of enforcements. However, in general, desired enforcements might not exist. Consider the following example.

Example 3.5. *Consider again the AF $F = (A, R)$ of Figure 4. We illustrate now three different sources for the impossibility of enforcements.*

1. *Assume we aim to strictly ad-enforce $E = \{b, d\}$ under normal expansions. While non-strict enforcement of E was possible (cf. Example 3.4), strict enforcement is impossible under normal expansions. The intuition is that $\{b, d\}$ is not admissible in the original AF F (the attack from c to d is not defended) and this fact remains true in any normal expansion G of F. The reason is simply that any new attack in G involves at least one new argument and thus, E can not defend d in G. However, E can be strictly enforced when allowing arbitrary expansions, e.g. adding a defending attack (b, c) is an option.*

2. *Another reason for impossibility of enforcement occurs when considering enforcement of sets like $\{a, b\}$ under any semantics σ that preserves conflict-freeness, i.e. $\sigma \subseteq cf$. The reason is that $\{a, b\}$ is conflicting in F and thus, it remains conflicting regardless the considered type of expansion.*

3. *Even if the set E to be enforced is conflict-free and defends all its elements, enforcement is, under specific semantics, not always possible. Consider the aim to strictly co-enforce $E = \{c\}$ under weak expansions. In F the singleton E is not complete since it defends a and $a \notin E$. Now, weak expansions do not raise new attacks onto existing arguments which implies that former defense relations survive. Thus, for any weak expansion G of F we have E still defends a preventing it from being complete in G.*

The previous observations have been firstly formalized in [17, Proposition 1] and later considered further in [43, Proposition 1]. In the following we recall some results and generalize them to other semantics considered in this article.

Proposition 3.6. *Given an AF $F = (A, R)$ and $E \subseteq A$.*

- *If $E \notin ad(F)$ and $\sigma \subseteq ad$, then there is no AF G strictly σ-enforcing E under normal expansions.*
- *If $E \notin cf(F)$ and $\sigma \subseteq cf$, then there is no AF G (non-)strictly σ-enforcing E under arbitrary expansions.*
- *If E does not contain all defended arguments in F and $\sigma \subseteq co$, then there is no AF G that strictly σ-enforcing E under weak expansions.*
- *If $\sigma \in \{ad, cf, na, stb\}$ and $E \notin \sigma(F)$, then there is no AF G strictly σ-enforcing E under normal expansions.*

Despite several cases being impossible to enforce, there are interesting conditions under which an enforcement is always possible. As an illustration, consider the following example.

Example 3.7. *Say we desire to non-strictly ad-enforce $E = \{b, d\}$ under strong expansions. This means, we want E to be a strict subset of an admissible extension of the expanded framework. An example AF G ad-enforcing $\{b, d\}$ is shown in Figure 5. Here the new argument e is added which defends both b and d. Since $\{e\}$ is admissible in G we obtain via the famous Fundamental Lemma [54, Lemma 10] that $\{e, b, d\}$ is admissible as desired.*

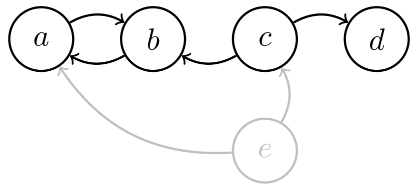


Figure 5: AF from Example 3.7

The observation from the example was generalized to further semantics [17, Theorem 4]. The main construction method is to extend the initial framework with a new argument which attacks all *undesired* arguments. We extend the already proven theorem to all semantics considered in this article.

Theorem 3.8. *Given an AF F, a desired set* $E \in cf(F)$ *and a semantics* $\sigma \in \{ad, stb, pr, co, gr, id\, sst, eg, na, stg\}$*. There is a strong expansion G of F non-strictly* σ*-enforcing E.*

Since strong expansions are particular cases of normal expansions as well as arbitrary expansions, we may state the following corollary.

Corollary 3.9. *Given an AF F, a desired set* $E \in cf(F)$ *and a semantics* $\sigma \in \{ad, stb, pr, co, gr, id\, sst, eg, na, stg\}$*. There are arbitrary as well as normal expansions G of F non-strictly* σ*-enforcing E.*

What about local expansions? Is it possible to (non-)strictly enforce a desired set E with local manipulations only? For most of the existing semantics we may act as follows: given the conflict-freeness of E we attack all remaining arguments first (this is sufficient for $\sigma \in \{ad, stb, pr, co, sst, na, stg\}$) and secondly, add self-loops to the remaining arguments (we additionally cover $\sigma \in \{id, eg\}$).

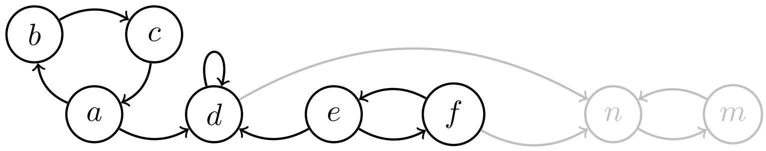
Theorem 3.10. *Given an AF F, a desired set* $E \in cf(F)$ *and a semantics* $\sigma \in \{ad, stb, pr, co, id\, sst, eg, na, stg\}$*. There is a local expansion G of F strictly* σ*-enforcing E.*

Note that grounded semantics is not included since it requires unattacked arguments which can not be “produced" with the help of local expansions. However, if there is an unattacked argument in the desired set E, then this unattacked argument can be used to attack all the arguments outside the directed set, leading to the strict gr-enforcement of E. Any unattacked argument in the AF can have a similar role for non-strict enforcement.

Theorem 3.11. *Given an AF F and a desired set* $E \in cf(F)$*, if there is an unattacked argument* $a \in E$ *(respectively* $a \in A$*), then there is a local expansion G of F strictly (respectively non-strictly) enforcing E under the grounded semantics.*

Let us turn now to a different aspect of enforcing, namely *how* exactly existing σ-extensions may change when expanding an AF. In general, the change is very much non-monotone: this means, arguments accepted earlier may become unaccepted, others become accepted; the number of extensions may shrink or increase, depending on the new arguments. For instance, it is easy to verify that we obtain a total collapse of stable extensions if we revise an AF by adding a self-defeating argument. Nevertheless, there are a few exceptions as illustrated in the following example taken from [15, Example 3.11]

Example 3.12. *Consider the weak expansion G of F as depicted below. In Example 2.3 we already observed that $pr(F) = \{\{e\}, \{f\}\} = \{E_1, E_2\}$.*



For the weak expansion G we find $pr(G) = \{E_1 \cup \{n\}, E_1 \cup \{m\}, E_2 \cup \{m\}\}$. Consequently, the following interrelations hold:

1. *the number of extensions increased*
2. *every old belief set is contained in a new one*
3. *every new belief set is the union of an old one and a new argument*

The previous example contrasts with the general observation that adding new arguments and attacks may change the outcome of an AF in a nonmonotonic fashion. Such a behaviour allows for reusing already computed extensions and has useful implications w.r.t. justification states. The following theorem [15, Theorem 3.2] shows that the class of weak expansions and semantics satisfying the directionality principle guarantee monotonic evolvements. Roughly speaking, the directionality criterion captures the idea that the evaluation of a certain argument should only be affected by its attackers and the attackers of its attackers and so on [9].

Theorem 3.13. *Given an AF $F = (A, R)$ and a semantics σ satisfying directionality, then for all weak expansions $G = (B, S)$ of F we have:*

1. $|\sigma(F)| \leq |\sigma(G)|$, *(cardinality)*
2. $\forall E \in \sigma(F)\ \exists E' \in \sigma(G)\ \exists C \subseteq B \setminus A,$ *s.t.* $E' = E \cup C$ *and* *(subset)*
3. $\forall E' \in \sigma(G)\ \exists E \in \sigma(F)\ \exists C \subseteq B \setminus A,$ *s.t.* $E' = E \cup C$. *(representation)*

It is well-known that admissible, complete, preferred, grounded and ideal semantics satisfy directionality (cf. [83] for an overview). Having the above theorem at hand we obtain the following relations regarding acceptance modes stating that credulously, sceptically as well as covered accepted sets persist.

Proposition 3.14. *Given an AF $F = (A, R)$ and $\sigma \in \{ad, co, pr, gr, id\}$. For any weak expansions G of F we have:*

- $cred(F, \sigma) \subseteq cred(G, \sigma)$,
- $scep(F, \sigma) \subseteq scep(G, \sigma)$ *and*
- $cov(F, ad) \subseteq cov(G, ad)$

3.2.2 Attack-based enforcement: Argument-fixed and Local Expansion-based Enforcement

We now turn to extension enforcement under a different kind of modifications to a given AF. In contrast to the previous section on expansion-based enforcement where expansion of the set of arguments and attacks, under certain conditions, was presented, we here look at changes that do not modify the set of arguments, but exclusively focus on updates of the attack structure.

Definition 3.15. *Let $F = (A, R)$ be an AF. We say that G is a local update of F, denoted by $F \asymp_L G$, if there is an AF G' such that $F \preceq_L G'$ and $G' \succeq_{LD} G$.*

In words, an AF $G = (A_G, R_G)$ is a local update of $F = (A_F, R_F)$ if there is an intermediate AF $G' = (A_{G'}, R_{G'})$ that is a local expansion of F (i.e., $A_{G'} = A_F$ and $R_F \subseteq R_{G'}$) and G is a local deletion of G' (i.e., $A_{G'} = A_G$ and $R_{G'} \supseteq R_G$). Put differently, G is a local update of F if the set of arguments stays the same, i.e., $A_G = A_F$, and the attack structure was changed arbitrarily: $R_G = (R_F \setminus R) \cup R'$ for some $R, R' \subseteq A_F \times A_F$.

In this section we consider extension enforcement under local updates [43]. An intuition of a local update is that the arguments are unmodified, but some new attacks are revealed (e.g., in presence of new information), and some attacks are disputed and discarded (e.g., due to the defeasibility of attacks). Modifying the attacks between existing arguments can also be seen as an update of the preferences between arguments [2].[3]

Example 3.16. *Let us look at the same AF F from the preceding section that we used to exemplify expansion-based enforcement. We recall this AF in Figure 6a.*

We begin with looking at enforcement of the set $\{b, d\}$. Say, we desire to have this set of arguments being part of an admissible extension. In F the set $\{b, d\}$ is conflict-free but not admissible: the attack from c onto both b and d is not countered. A local update, in fact a local expansion, that enforces $\{b, d\}$ to be part of an admissible extension is shown in Figure 6b. An attack from b to c suffices to have $\{b, d\}$ defend both b and d.

[3]Recall that in preference-based argumentation, the "success" of an attack (a, b) depends on the fact that b is not preferred to a.

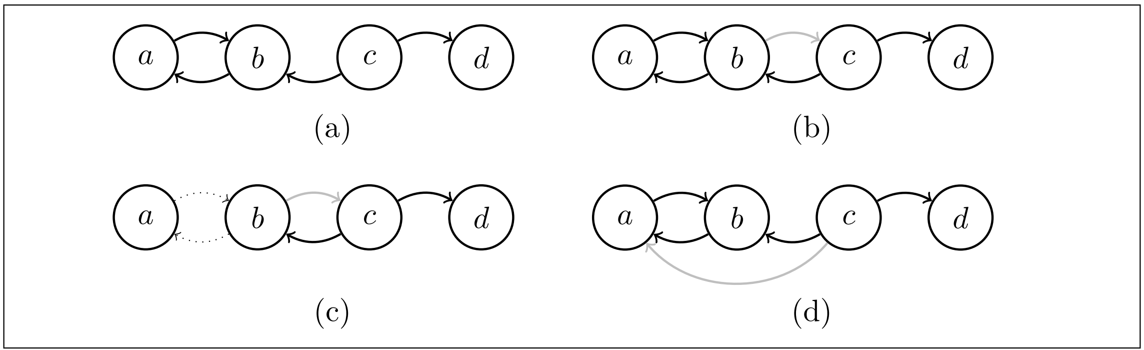


Figure 6: Enforcement by local updates

A different case is exhibited by aiming to have $\{a, b\}$ *being admissible: this set neither is conflict-free nor defends its arguments. A possible local update is shown in Figure 6c that enforces* $\{a, b\}$ *to be exactly an admissible extension, i.e., realizes strict extension enforcement under local updates and admissibility. Here, the conflicts between* a *and* b *are removed, to ensure conflict-freeness, and the attack from* b *to* c *is added, to ensure defense.*

Finally, consider strict enforcement of $\{c\}$ *under complete semantics. The set* $\{c\}$ *is admissible, yet defends* a *in* F*. A possible local update (local expansion) is shown in Figure 6d. Here one attack from* c *to* a *ensures that* $\{c\}$ *does not defend* a*.*

Inspection of the preceding example reveals that several impossible cases, when requiring certain expansions (see previous section), are, in fact, possible under local updates. This is no coincidence: enforcement under local updates is possible for all main semantics of AFs: if $E \neq \emptyset$ is to be enforced, for a given AF $F = (A, R)$ there is the (trivial) local update $G = (A, R')$ with $R' = \{(a, b) \mid a \in E, b \in A \setminus E\}$ (i.e., in G, every argument in E is non-attacked, and every argument in $A \setminus E$ is attacked by all arguments in E). We have $E \in gr(G)$, and since the graph structure of G is acyclic[4], most semantics coincide with the grounded semantics.

This observation is formalized next [43, Proposition 4].

Proposition 3.17. *Let* $F = (A, R)$ *be an AF and* $E \subseteq A$ *be a non-empty set of arguments. There exists a local update* G *that enforces* E *(non-)strictly to be (part of) a* σ*-extension, for all* σ *considered in this chapter.*

[4] In the case of finite AFs, acyclicity corresponds to the well-foundedness property defined by [54], which implies the coincidence of grounded, stable, preferred and complete semantics. We also refer the reader to [7] for more details on this topic.

Obviously, when $E = \emptyset$, it can always be non-strictly enforced with local updates, since E is included in any set of arguments. It is also the case that E can be strictly enforced with local update.[5] Indeed, for a given AF $F = (A, R)$ we can define the (trivial) local update $G = (A, R')$ with $R' = \{(a, a) \mid a \in A\}$ (i.e., in G, every argument is self-attacking). In this case, the empty set is the only conflict-free set, and thus the only extension for most semantics.

We have seen that enforcing a set of arguments with local updates is possible in general. Both the addition and the removal of attacks are necessary for this results. Indeed, if only local expansions are possible (i.e. removing attacks is not permitted), then a conflicting set E cannot be enforced under any semantics that requires conflict-freeness. Similarly, local deletions are not sufficient for strictly enforcing a set of arguments in all cases. As a matter of example, let us consider again the AF $F = (A, R)$ given at Figure 6a. The set $\{c\}$ cannot be enforced as a stable extensions by only deleting attacks: initially $\{c\}^{\oplus} = \{b, c, d\} \neq A$, and removing attacks cannot add arguments to the range of $\{c\}$.

3.2.3 Extension Enforcement and Minimal Change

Minimal change is an important topic in other domains of artificial intelligence, like belief change [1; 63]. In the context of extension enforcement, the question asked is "how much effort will it cost to perform the enforcement?". This effort is defined by [13] as the number of attacks that are modified (i.e. either added or removed). Formally,

Definition 3.18. *Given $F = (A, R)$ and $G = (A', R')$, the distance between F and G is $d(F, G) = |(R \setminus R') \cup (R' \setminus R)|$.*

In general, there may be several ways to enforce an extension, even for a fixed type of modification. In that case, minimal change enforcement consists in choosing one result that minimizes the distance d between the initial AF and the new one.

Example 3.19. *Figure 7 presents two examples of strong expansions of an AF $F = (A, R)$, with $A = \{a, b, c, d, e\}$ and $R = \{(b, a), (c, a), (d, b), (d, c), (e, d)\}$. This AF has a single stable extension: $stb(F) = \{\{b, c, e\}\}$. Both expansions succeed in non-strictly enforcing the set $\{a\}$ as a stable extension. However, we observe a difference in the number of attacks that have been added. The first one, F_1 (on the left side), adds two attacks, one from the new argument f_1 to c, and another one from f_2 to b; it has a single stable extension $stb(F_1) = \{\{a, e, f_1, f_2\}\}$. The second*

[5]Except for the stable semantics, since the empty set can never be a stable extension of a non-empty AF.

expansion, F_2 (on the right side), adds a single attack (f_3, e), and it also has a single stable extension: $stb(F_2) = \{\{a, d, f_3\}\}$. With $d(F, F_1) = 2$ and $d(F, F_2) = 1$, F_2 seems to be a more desirable result.

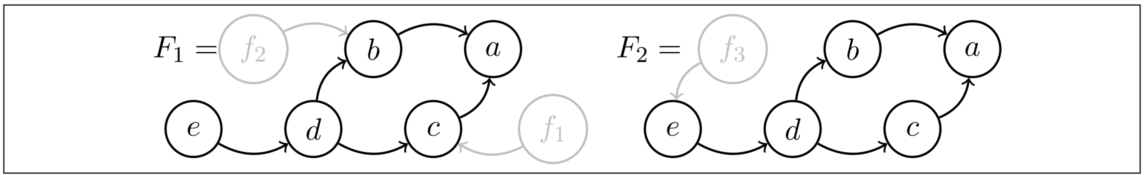


Figure 7: An example of (non-)minimal change

The question of minimal change in enforcement is studied in [13]. More specifically, it concerns the minimal change in non-strict enforcement based on normal expansions, as well as the special cases of strong and weak expansions. To do so, he defines the notion of characteristic of a set of arguments S, with respect to an AF F and a modification type $M \subseteq \mathcal{F} \times \mathcal{F}$. This characteristic corresponds to the minimal distance between F and an AF G such that S is included in an extension of G, and G is a possible result for the enforcement (i.e. $(F, G) \in M$). Strict enforcement can be considered as well [52].

Definition 3.20. *Given a semantics σ, a modification type $M \subseteq \mathcal{F} \times \mathcal{F}$, $x \in \{s, ns\}$ meaning strict or non-strict, and an AF $F = (A, R)$, the (σ, M, x)-characteristic of a set $S \subseteq A$ is:*

$$N^{F,x}_{\sigma,M}(S) = \begin{cases} 0 & \textit{if} \quad x = s, S \in \sigma(F) \\ 0 & \textit{if} \quad x = ns, \exists S' \in \sigma(F) \ \textit{s.t.} \ S \subseteq S' \\ k & \textit{if} \quad k = \min(\{d(F, G) \mid (F, G) \in M, N^{G,x}_{\sigma,M}(S) = 0\}) \\ +\infty & \quad \textit{otherwise} \end{cases}$$

Intuitively, the characteristic of a set of arguments S is 0 if this set is already (included in) an extension, k if k is the minimal distance between the initial AF and some AF that enforces S, and $+\infty$ if S cannot be enforced (under the the specified semantics and modification type).

Then, [13] introduces the notion of *value function*, that gives a constructive definition of how to compute the characteristic in a finite number of steps, based on properties of the initial AF. We use $V^{F,x}_{\sigma,M}(S)$ to denote this value function.

We start with the case of non-strict enforcement under weak expansion. Baumann shows that for most semantics, either the set S is already included in an extension, or it is impossible to enforce it with a weak expansion [13, Theorem 6]. Formally,

Proposition 3.21. *For $\sigma \in \{stb, ad\}$ a semantics, $F = (A, R)$ and AF and $S \subseteq A$ a set of arguments, the value function for non-strict enforcement under weak expansion and the semantics σ is*

$$V^{F,ns}_{\sigma,\preceq_W}(S) = \begin{cases} 0 & \textit{if} \ \ \exists S' \in \sigma(F) \ \textit{s.t.} \ S \subseteq S' \\ +\infty & \textit{otherwise} \end{cases}$$

Then, $N^{F,ns}_{stb,\preceq_W}(S) = V^{F,ns}_{stb,\preceq_W}(S)$ and $N^{F,ns}_{\sigma,\preceq_W}(S) = V^{F,ns}_{ad,\preceq_W}(S)$ for $\sigma \in \{ad, co, pr\}$.

Now, we turn to (non-strict) enforcement under strong expansion., i.e. we focus on defining $V^{F,ns}_{\sigma,\preceq_S}(S)$. This case is slightly more involved than the previous one, and it requires additional definitions.

Definition 3.22. *Given $F = (A, R)$ an AF and $X \in cf(F)$,*

- $ad(F, X) = X^{\ominus} \setminus X^{\oplus}$;
- $stb(F, X) = A \setminus X^{\oplus}$.

Intuitively, these sets correspond to the arguments that should be defeated in order to make X an admissible (respectively stable) extension of F. They can be used to define the value function for enforcement under strong expansion, for $\sigma \in \{stb, ad\}$. Interestingly, these value functions can be used also for enforcing a set of arguments under normal expansion or general expansions, as stated by [13, Theorem 9].

Proposition 3.23. *For $\sigma \in \{stb, ad\}$ a semantics, $F = (A, R)$ and AF and $S \subseteq A$ a set of arguments, the value function for non-strict enforcement under strong expansion and the semantics σ is*

$$V^{F,ns}_{\sigma,\preceq_S}(S) = \min(\{|\sigma(F, S')| \mid S \subseteq S' \textit{ and } S' \in cf(F)\})$$

Then, $N^{F,ns}_{stb,M}(S) = V^{F,ns}_{stb,\preceq_S}(S)$ and $N^{F,ns}_{\sigma,M}(S) = V^{F,ns}_{\sigma,\preceq_S}(S)$ hold for $\sigma \in \{ad, co, pr\}$ and $M \in \{\preceq_E, \preceq_N, \preceq_S\}$.

This means that authorizing more kinds of modifications than the addition of strong arguments is useless regarding the issue of minimal change.

Then, an interesting result [13, Proposition 11] states that enforcement is always possible if arbitrary updates are permitted, i.e. attacks can also be deleted (contrary to expansions, where attacks can only be added).

Proposition 3.24. *For $\sigma \in \{stb, sst, pr, co, ad\}$ and any $F = (A, R)$,*

$$N^{F,ns}_{\sigma,U}(S) \leq |R \cap (S \times S)| + |A \setminus S|$$

Intuitively, it says that we can enforce S as (a subset of) an extension by making it conflict-free (i.e. removing the attacks in $R \cap (S \times S)$) and attacking every argument that is not in S (i.e. adding attacks from fresh arguments to arguments in $A \setminus S$). This finite upper bound guarantees that non-strict enforcement under arbitrary updates is always possible. But a more precise evaluation of the characteristics is given by this value function [13, Theorem 12]:

Proposition 3.25. *For $\sigma \in \{stb, ad\}$ a semantics, $F = (A, R)$ and AF and $S \subseteq A$ a set of arguments, the value function for non-strict enforcement under arbitrary updates and the semantics σ is*

$$V_{\sigma,U}^{F,ns}(S) = \min(\{|R \cap (S' \times S')| + |\sigma(F, S')| \mid S \subseteq S' \subseteq A\}$$

with $ad(F, S')$ and $stb(F, S')$ as in Definition 3.22. Then, $N_{stb,U}^{F,ns}(S) = V_{stb,U}^{F,ns}(S)$ and $N_{\sigma,U}^{F,ns}(S) = V_{\sigma,U}^{F,ns}(S)$ hold for $\sigma \in \{pr, co, ad\}$.

Finally, [52] presents characteristics for enforcement under local updates, i.e. when the set of arguments has to remain the same, but attacks between them can be added or deleted. The results are reminiscent of the ones described in this section.

3.2.4 Semantics-based Enforcement

Extension enforcement is usually defined as an operation where the target semantics is given as an input. We call it conservative enforcement when the target semantics is the same as the initial semantics, and liberal enforcement otherwise. On the contrary, [52] proposes to generalize enforcement, by enhancing operators with a set Σ of possible target semantics. Then, the chosen semantics is the one that allows to enforce the set of arguments with minimal change on the graph. More formally:

Definition 3.26. *For $F = (A, R)$ an AF, $S \subseteq A$ the set of arguments to be enforced and Σ a set of semantics, a strict (resp. non-strict) enforcement of S in F under a given modification type $M \subseteq \mathcal{F} \times \mathcal{F}$, is a pair (G, σ') such that*

1. *$(F, G) \in M$;*
2. *$\sigma' \in \Sigma$ and $S \in \sigma'(G)$ (resp. $S \subseteq S' \in \sigma'(G)$);*
3. *$\forall \sigma'' \in \Sigma$, $V_{\sigma',M}^{F,x}(S) \leq V_{\sigma'',M}^{F,x}(S)$ (with $x \in \{s, ns\}$).*

This means that the new semantics is chosen in a way that guarantees that the change on the graph is minimal. Since the characteristics can be the same for several semantics σ', additional criteria can be used in order to select the new semantics, like the distance between σ' and the initial semantics σ [51].

Finally, we already mentioned that [17, Section 3.1] discusses the tool of changing semantics in order to enforce a desired set. The authors presented two involved impossibility theorems specifying properties of initial extensions and desired sets, initial and target semantics as well as the considered type of structural change. Regarding the semantic change we have that possible target semantics were restricted to semantics satisfying well-known abstract criteria like admissibility or reinstatement (cf. [83] for an exhaustive overview). The mentioned theorems show either limitations for exchanging accepted arguments with formerly unaccepted ones (under normal expansions) or limitations for eliminating arguments of existing extensions (under weak expansions).

3.3 Complexity and Algorithms

We review complexity of enforcement problems, in particular expansion-based enforcement, and enforcement based on local updates [85; 43].

In several cases enforcement is, computationally speaking, straightforward if the task consists in checking whether there exists a modified AF that enforces a set of arguments under certain parameters. For instance, extension enforcement under normal expansions for admissible semantics is always possible if the set E to enforce is conflict-free in the given AF (see Section 3.2.1). That is why we look at extension enforcement that aims at minimizing the change induced by an enforcing AF. Concretely, given an AF $F = (A, R)$ we aim at finding an enforcing AF $G = (A', R')$ such that the distance $d(F, G)$ between them is minimal (see Definition 3.18).

Another important aspect for expansion-based enforcement is how many arguments shall be added. That is, if $G = (A', R')$ is an expansion of $F = (A, R)$, how to confine $|A'| - |A|$? This is important from a computational perspective, since allowing for unbounded expansions may complicate computation. We consider here only bounded expansions.

We define the computational problems next, for extension enforcement under bounded expansions and local updates. For local updates no bound is needed, since if the number of arguments $|A|$ does not change, the number of modifications to R is bounded quadratically by $|A|$.

For studying complexity of problems that are inherently optimization problems, such as enforcement when the goal is to find an enforcing AF with a minimum number of modifications to the attack structure, there are several ways to formally approach such problems. One standard way to reveal inherent complexities of optimization problems is to consider a natural decision variant: for a given integer $k \geq 0$, we ask whether there is an enforcing AF with at most k many modifications. We note that another way to study complexity of optimization problems is to consider functional problems instead of decision problems, which is an approach that may give more detailed complexity results (see, e.g., [64]). However currently no such analysis was carried out for enforcement.

First, we define a decision problem for extension enforcement under bounded expansions.

Extension enforcement under bounded expansions
Instance: an AF $F = (A, R)$, $E \subseteq A$, set A', integer $k \geq 0$, and a semantics σ.
Question: Does there exist an expansion $G = (A \cup A', R')$ of F such that $\exists E' \in \sigma(G)$ with $E \subseteq E'$ and $d(F, G) \leq k$?

In more words, given an AF F, a set $E \subseteq A$ of arguments to enforce, a set of arguments A', an integer $k \geq 0$ and a semantics σ, the task is to decide whether there exists an expansion G of F that enforces E non-strictly under σ, and, moreover, makes at most k many modifications to the attack structure. Note that the expansion G is bounded in the sense that the expanded arguments are already given beforehand, i.e., G has $A \cup A'$ as its arguments. The above definition gives a decision problem for non-strict enforcement. As before, we define strict enforcement analogously by replacing $\exists E' \in \sigma(G)$ and $E \subseteq E'$ with $E \in \sigma(G)$.

Next, we look at a decision problem variant for extension enforcement under local updates.

Extension enforcement under local updates
Instance: an AF $F = (A, R)$, $E \subseteq A$, integer $k \geq 0$, and a semantics σ.
Question: Does there exist a local update $G = (A, R')$ of F such that $\exists E' \in \sigma(G)$ with $E \subseteq E'$ and $d(F, G) \leq k$?

Strict enforcement is again defined as above.

We consider as fragments of these two enforcement problems those sub problems where a semantics is fixed, i.e., extension enforcement under bounded expansions (local updates) under a specific semantics σ, instead of having σ as part of the instance.

Finally, before delving into complexity results from the literature, we provide background on complexity classes used here, and related problems useful to understanding complexity of enforcement. For thorough introductions to computational complexity see, e.g., [3; 77]. We assume that the reader is familiar with concepts like complexity classes, reductions, completeness, and oracles. Complexity class P is composed of all decision problems which can be decided in polynomial time by a deterministic algorithm. Class NP contains all decision problems that can be decided by a non-deterministic polynomial time algorithm. Class coNP contains all problems that are complementary to a problem in NP. Class Σ_2^P contains all decision problems which can be decided via a non-deterministic polynomial time algorithm that can access an NP oracle. Class Π_2^P contains all problems that are complementary to some problem in Σ_2^P.

Two reasoning tasks on AFs in a static, i.e., non-dynamic setting, are useful to understand the complexity of enforcement. The first one is usually referred to as the Verification problem.

Verification
Instance: an AF $F = (A, R)$, $E \subseteq A$, and a semantics σ.
Question: Does $E \in \sigma(F)$ hold?

That is, the task is to check whether a given set E is a σ-extension. Another useful problem is credulous acceptance of arguments in AFs.

Credulous acceptance
Instance: an AF $F = (A, R)$, $a \in A$, and a semantics σ.
Question: Does $\{a\} \in cred(F, \sigma)$ hold?

In words, an argument is credulously accepted in case there is a σ-extension of a given AF that contains the queried argument.

Complexity of verification and credulous acceptance was established; we summarize complexity results for the main semantics in Table 2.

semantics σ	verification	credulous acceptance
cf	in P	in P
ad	in P	NP-complete
co	in P	NP-complete
stb	in P	NP-complete
pr	coNP-complete	NP-complete

Table 2: Complexity of verification and credulous reasoning in AFs (for an overview see [57])

3.3.1 Complexity of Enforcement

We illustrate two ways of showing complexity bounds that turn out to be tight in many, but not all, cases.

For an upper bound (i.e. membership in a complexity class), consider the following non-deterministic algorithm (sketch) given an AF F, a set E to enforce, and a semantics σ:

1. non-deterministically construct an AF $G = (A', R')$ that is a bounded expansion (or local update) of F;
2. non-deterministically construct an $E' \subseteq A'$ (for non-strict enforcement only); and
3. verify whether $E' \in \sigma(G)$ and $E \subseteq E'$ (for non-strict enforcement) or $E \in \sigma(G)$ (for strict enforcement).

In case the last step succeeds, then it holds that G enforces E to be a σ-extensions (non-)strictly. As can be seen from this algorithm sketch, a complexity upper bound can be derived from the complexity of the verification problem under σ. Take $\sigma = ad$, i.e., the verification problem under admissibility which is polynomial-time decidable. It follows that extension enforcement under bounded expansions (resp. local updates) is in NP under admissibility. The reason is that the above algorithm sketch directly witnesses membership in NP: one (resp. two) non-deterministic construction(s) and a check in polynomial time show membership for $\sigma = ad$. In the non-deterministic construction of the above algorithm the bound on the expansion is crucial, otherwise a non-bounded, and thus potentially non-polynomially bounded, structure would be constructed. However, this does not imply that enforcement under non-bounded expansions requires large expansions.

There is a similar approach to show lower bounds. Here we distinguish more between strict and non-strict variants. In particular, extension enforcement under bounded expansions (resp. local updates) and σ is C-hard

- if verification under σ is C-hard and the enforcement variant is strict; or
- if credulous acceptance under σ is C-hard and the enforcement variant is non-strict.

The underlying reason is as follows. One can reduce the verification problem to strict extension enforcement and the credulous acceptance problem to non-strict extension enforcement.

For the verification problem under σ, i.e., given an AF F and a set E, consider the extension strict enforcement problem under σ with F, E, and $k = 0$ as input (and $A' = \emptyset$ for expansion-based). Then we are not allowed to make modifications to F, and, therefore, F enforces E to be a σ-extension iff $E \in \sigma(F)$ iff this is a positive instance of the verification problem.

Similarly, the credulous acceptance problem under σ, with F and an argument a as input, is reduced to an instance of non-strict extension enforcement with input F, $E = \{a\}$, and $k = 0$ (and again $A' = \emptyset$). It follows that F enforces E non-strictly if there is an $E' \supseteq E$ with $E' \in \sigma(F)$, implying a positive instance of the credulous acceptance problem.

In several cases the two approaches to show upper and lower bounds yield tight bounds. However, there are notable exceptions.

Let us look first at results obtained for enforcement under bounded expansions, see Table 3. In this case only the non-strict variant was studied [85]. It can be observed that the complexity of this enforcement variant matches complexity of credulous reasoning in static AFs, i.e., the above approaches to show complexity bounds directly result in tight bounds. We remark that complexity of enforcement under conflict-free sets was not presented in [85], however it can be straightforwardly obtained: if the set is conflict-free then enforcement is trivial (and can be checked in polynomial time by scanning the input AF), otherwise, if the given set to enforce is conflicting, no expansion can remove such conflicts, and enforcement is impossible. Since (by definition) any conflict-free set is included in some naive extension, this result also holds for $\sigma = na$.

semantics σ	non-strict
cf	in P
na	in P
ad	NP-c
co	NP-c
stb	NP-c
pr	NP-c

Table 3: Complexity of non-strict extension enforcement under bounded expansions [85]

Let us turn to complexity of enforcement under local update [85; 43], summarized in Table 4. We see that complexity of non-strict enforcement, again, has the same complexity as credulous reasoning in static AFs, except for grounded semantics. Before discussing grounded semantics, let us turn to strict enforcement first.

To some extend surprising are the results for strict extension enforcement, which diverge from non-strict enforcement. For instance, for both admissible and stable semantics strict extension enforcement under local expansions is decidable in polynomial time. The underlying reason is that if E is to be an admissible set (a stable extension), then all conflicts inside the set have to be removed, and each attack

semantics σ	strict	non-strict
cf	in P	in P
na	in P	in P
ad	in P	NP-c
co	NP-c	NP-c
gr	NP-c	NP-c
stb	in P	NP-c
pr	Σ_2^P-c	NP-c
sst	Σ_2^P-c	Σ_2^P-c
stg	coNP-hard and in Σ_2^P	Σ_2^P-c

Table 4: Complexity of extension enforcement under local updates [85; 43]

from outside countered (each argument outside attacked). The latter means that one can choose an argument inside E to counter non-attacked attackers (to achieve defense) or remove the incoming attack. In both cases, it is sufficient to make at least one modification, however one modification is sufficient: adding an attack to counter an attacker (removing an incoming attack might not be sufficient if there are more incoming attacks). For stable semantics, similarly, one attack on unattacked arguments outside E is both necessary and sufficient, only the origin in E is flexible. Overall, this procedure sketches a polynomial-time deterministic algorithm (one can impose an ordering on arguments to make the choice of attacking arguments deterministic).

Finally, let us look at grounded semantics, for which NP-completeness was established for both non-strict and strict extension enforcement under local updates and complete semantics for the strict variant. Recall that both verification and credulous acceptance under grounded semantics is in P, and also verification for complete semantics is in P (Table 2). This means, the lower bounds established by the algorithms above do not result in tight bounds. The intuition behind this "complexity jump" for the strict variant under complete and grounded semantics is that when enforcing some set of arguments E to be complete, one has to be careful about what E defends. That is, enforcing E to be admissible is not the underlying reason for NP hardness, but to avoid having arguments defended that one desires to avoid being defended (as specified by strict enforcement, nothing outside the set E may be defended by E). In brief, addition or removal of attacks can make E admissible, but implying further arguments being defended. Finding an optimal assignment that accomplishes both having E admissible and nothing outside E being defended by E faces non-deterministic choices. However, the hardness construction to show NP-hardness is somewhat involved.

Finally, for grounded semantics and non-strict enforcement, the intuition for NP-completeness is a bit more direct: there could be a place in the AF to modify such that the grounded extension is significantly expanded and includes the desired E. However, choosing an adequate place in such a way is not direct to find.

Further semantics have been analyzed in [85].

3.3.2 Declarative Algorithms

Main approaches to compute optimal enforcing for AFs rely on declarative programming paradigms based on constraints, in particular maximum Satisfiability (MaxSAT) [68], answer set programming (ASP) [69; 58], and pseudo Boolean optimization (particularly integer linear programming [80]).

We present here some of the main ideas for algorithmic approaches to extension enforcement, focusing on the MaxSAT approach [85]. Enforcement via pseudo Boolean optimization is presented in [43], and via ASP in [74]. Systems using the MaxSAT approach are presented in [73; 43]. We present here encodings and an algorithm for some semantics, for further semantics and details we refer to the original papers.

We briefly recall background on MaxSAT. A literal is either a positive Boolean variable x or a negated Boolean variable $\neg x$. A clause is a disjunction of literals $l_1 \vee \cdots \vee l_n$ and a propositional formula is in conjunctive normal form (CNF) if the formula $\pi = c_1 \wedge \cdots \wedge c_m$ is a conjunction of clauses. Whenever convenient, we will view clauses as a set of literals and a formula in CNF as a set of clauses.

A truth assignment τ assigns either true (1) or false (0) to the Boolean variables. As usual, a truth assignment τ satisfies a variable x if $\tau(x) = 1$. Satisfaction is extended in the usual way to compound formulas, e.g., τ satisfies a literal l if $\tau(x) = 1$ and $l = x$ or $\tau(x) = 0$ and $l = \neg x$. A clause is satisfied by τ if at least one literal of the clause is satisfied, and a formula in CNF is satisfied if each clause is satisfied.

An instance of the partial MaxSAT problem is a pair $\phi = (\phi_h, \phi_s)$ with both ϕ_h and ϕ_s Boolean formulas in CNF (sets of clauses). The former is the set of hard clauses, while the latter is the set of soft clauses. A truth assignment τ is a solution to the partial MaxSAT instance if τ satisfies ϕ_h (the hard clauses). The cost of τ w.r.t. the instance ϕ is $\texttt{cost}(\phi, \tau) = \sum_{c \in \phi_s} 1 - \tau(c)$, i.e., the number of soft clauses not satisfied. A solution τ to ϕ is optimal if there is no solution τ' to ϕ with $\texttt{cost}(\phi, \tau') < \texttt{cost}(\phi, \tau)$. We refer to partial MaxSAT simply as MaxSAT.

We focus on an illustration of a MaxSAT approach to extension enforcement on the variant with local updates. Encoding extension enforcement under local updates can be achieved by encoding an AF, possible modifications, and semantics in MaxSAT. Before delving into encoding this enforcement variant, we recall an encoding of admissible semantics of static AFs similar as in [26]. Given an AF $F = (A, R)$ we define

$$\phi_{cf}(F) = \bigwedge_{(a,b)\in R} \neg a \vee \neg b.$$

We use here arguments as Boolean variables and vice versa. Satisfying truth assignments of this formula correspond directly to conflict-free sets of F in the way that $E \in cf(F)$ iff $\tau(x) = 1$ for $x \in E$ and $\tau(y) = 0$ if $y \notin E$ satisfies ϕ_{cf}. Admissibility can be encoded as follows:

$$\phi_{ad}(F) = \phi_{cf}(F) \wedge \bigwedge_{a\in A} (a \rightarrow (\bigwedge_{(b,a)\in R} (\bigvee_{(c,b)\in R} c)))$$

In words, if an argument a is in an admissible set (true in a satisfying assignment) then for each attacker b at least one defender c must be part of the admissible set (true in the assignment), as well, which directly captures the definition of admissibility.

Non-strict extension enforcement under local updates can be encoded by including variables for attacks. We first focus on how to encode constraints for the semantics, which we encode as hard clauses ϕ_h. For notation, for the encodings of conflict-free sets and admissible extensions above we used ϕ, for enforcement formulas we use ψ.

$$\psi_{cf}(F) = \bigwedge_{a,b\in A} (r_{a,b} \rightarrow (\neg a \vee \neg b))$$

In words, a new variable $r_{a,b}$ for each pair of arguments a, b is introduced denoting whether there is an attack from a to b. That is, a truth assignment includes now an assignment on the attacks, as well.

Moving on to enforcement under admissibility, which we encode as

$$\psi_{ad}(F) = \psi_{cf}(F) \wedge \bigwedge_{a,b\in A} ((a \wedge r_{b,a}) \rightarrow \bigvee_{c\in A} (c \wedge r_{c,b})).$$

That is, if a is assigned to be true and there is an attack (b, a), then this attack has to be defended against, by some c and the corresponding attack (c, b).

Let $F = (A, R)$, $E \subseteq A$ be given as an instance of the non-strict extension enforcement problem under local updates and admissibility. Defining a MaxSAT instance

$$\phi = (\psi_{ad} \wedge \bigwedge_{a \in E} a, \phi_s(F))$$

with

$$\phi_s(F) = \bigwedge_{(a,b) \in R} r_{a,b} \wedge \bigwedge_{(a,b) \notin R, a,b \in A} \neg r_{a,b}$$

results in optimal truth assignments τ to ϕ corresponding to AFs locally updated from an original AF $F = (A, R)$ with a minimum number of modifications that enforce S to be part of an admissible set. To see this, any solution to ϕ satisfies the hard clauses, implying that a truth assignment satisfying $\psi_{ad} \wedge \bigwedge_{a \in E} a$ assigns to true all variables (arguments) in E and possibly more arguments, and further assigns some of the attacks to be true (present in a modified AF) such that the argument variables assigned to true form an admissible set, thus enforcing E.

For strict extension enforcement under local updates, more variables can be fixed, since the set E to be enforced must be exactly a σ-extension, not just be part of one. That is, we can focus on variables only for attacks, since the other variables can be fixed (true for argument variables in E and false otherwise, i.e. $\bigwedge_{a \notin E} \neg a$ can be added to the hard part of the MaxSAT instance).

For instance, strict enforcement under conflict-free sets can be encoded as follows.

$$\psi^s_{cf}(F) = \bigwedge_{a,b \in E} \neg r_{a,b}$$

In more words, there cannot be any attack in the set S to be enforced, all other attacks remain unconstrained for conflict-free sets. Admissible extensions can be encoded by

$$\psi^s_{ad}(F) = \psi^s_{cf}(F) \wedge \bigwedge_{a \in E} \bigwedge_{b \in A \setminus E} (r_{a,b} \rightarrow \bigvee_{c \in E} r_{c,b}).$$

The MaxSAT instance is complete by setting

$$\phi = (\psi^s_{ad}(F) \wedge \bigwedge_{a \in E} a \wedge \bigwedge_{a \in A \setminus E} \neg a, \phi_s(F)).$$

We proceed to algorithmic approaches for problems "beyond NP", e.g., strict extension enforcement under local updates and preferred semantics. One approach to such complex problems is to develop an algorithm that uses SAT solvers as subprocedures, and possibly calls a SAT solver multiple times (i.e., an iterative SAT-based procedure). We present an approach based (inspired by) the well-known CEGAR approach [40; 41] approach, where CEGAR stands for counterexample guided abstraction refinement. We remark that the term CEGAR is not used unambigously in the literature, and in some communities the term may refer to different concepts. Here, a CEGAR based algorithm works on an abstraction (approximation) of a solution space from which iteratively candidates are drawn. Importantly, due to the approximation, some solutions may be spurious. A SAT call determines whether a candidate is a solution or a spurious solution. In the latter case, the spurious solution is a "counterexample" which is used to refine the approximated solution space (removing as many as possible of the spurious solutions from the space) and a next candidate is produced, until a solution is reached.

For strict extension enforcement under local updates and preferred semantics, the solution space is approximated by considering initially strict extension enforcement under local updates and admissible or complete semantics. It holds that if an AF G enforces a set S strictly under preferred semantics, then G also enforces S strictly under admissible or complete semantics (since a preferred extension is complete and admissible). However, importantly, optimality is not guaranteed this way: an optimal solution to strict extension enforcement under local updates and preferred semantics might not be an optimal solution AF for admissible or complete semantics (since less modifications might be sufficient for admissible or complete semantics, but not for preferred semantics). Nevertheless, strict extension enforcement under complete or admissible semantics can act as an approximation. We focus for illustration on admissible semantics here.

The CEGAR-style algorithm is presented as Algorithm 1. When the loop is entered the first time, the MaxSAT call returns an optimal solution for strict extension enforcement under local updates and admissible semantics. To check whether the AF extractable from the truth assignment τ is a solution also under preferred semantics, we call a SAT solver to determine whether S is a preferred extension in the candidate AF. If so, we return this AF. Otherwise, we found a counterexample and the abstraction is refined. For correctness of the overall algorithm, it is important that a refinement does not remove (all) optimal solutions AFs for strict enforcement under local updates and preferred semantics. We refine here by removing the solution found in the MaxSAT call, i.e., by removing exactly τ from consideration when looking for the next candidate. This is a straightforward refinement. More sophisticated refinements are possible, but require care when designing them (e.g., in order not to violate correctness) [70]. For instance, in some cases refinements can be based on foundational results whether changes on AFs induce changes on semantics [31; 30].

The definitions for Algorithm 1 are as follows. From a truth assignment τ we can extract an AF by $\textsc{Extract}(A, \tau) = (A, R)$ with $R = \{(a, b) \mid \tau(r_{a,b}) = 1\}$. The

Algorithm 1 Strict extension enforcement under local updates and preferred semantics

$\varphi_h \leftarrow \psi'_{ad}(F)$
$\varphi_s \leftarrow \bigwedge_{(a,b)\in R} r_{a,b} \wedge \bigwedge_{(a,b)\notin R, a,b\in A} \neg r_{a,b}$
while true **do**
  $\tau \leftarrow$ MaxSAT(φ_h, φ_s)
  result $\leftarrow$ SAT(Check(τ, S))
  if *result* = *unsatisfiable* **then return** τ
  else $\phi_h \leftarrow \phi_h \wedge$ Refine(τ)
end while

formula

$$\textsc{Check}(\tau, E) = \phi_{ad}(F') \wedge \bigwedge_{a\in S} a \wedge \bigvee_{b\in A\setminus S} b$$

can be used for checking whether an admissible extension E is a preferred extension of the modified AF: we guess a superset and check admissibility by the above sub formulas. Refinement is specified via

$$\textsc{Refine}(\tau) = \bigvee_{(a,b)\in R'} \neg r_{a,b} \vee \bigvee_{(a,b)\in(A\times A)\setminus R'} r_{a,b}.$$

In words, we exclude in a subsequent search for a solution candidate exactly the currently found candidate AF.

Example 3.27. *Consider the AF F from Figure 8(a). That is, we have a chain of attacked arguments from a to b to c, and c attacks both d and e. Further, f is unattacked and does not attack an argument. The unique preferred extension of F is $\{a, c, f\}$. Say we want to strictly enforce $\{b, f\}$ to be exactly a preferred extension, and use Algorithm 1 in order to achieve that. Initially, we solve, via MaxSAT, strict enforcement under admissible semantics to have $\{b, f\}$ being admissible. Say the result is as shown in Figure 8(b), i.e., an attack from f to a is added, resulting in $\{b, d, e, f\}$ being the unique preferred extension, and $\{b, f\}$ being admissible. As the SAT solver call in the algorithm verifies, this AF candidate is not a solution, since, e.g., $\{b, d, e, f\}$ is an admissible superset of $\{b, f\}$, implying that $\{b, f\}$ is not preferred. We exclude, via the refinement step, this candidate AF, and call the MaxSAT solver again. Note that the hard clauses refute the previous candidate AF (i.e., any truth assignment simulating that AF does not satisfy the hard clauses).*

In the next steps of the algorithm, all AFs that enforce $\{b, f\}$ to be admissible are checked which make at most one modification (in the above simple refinement).

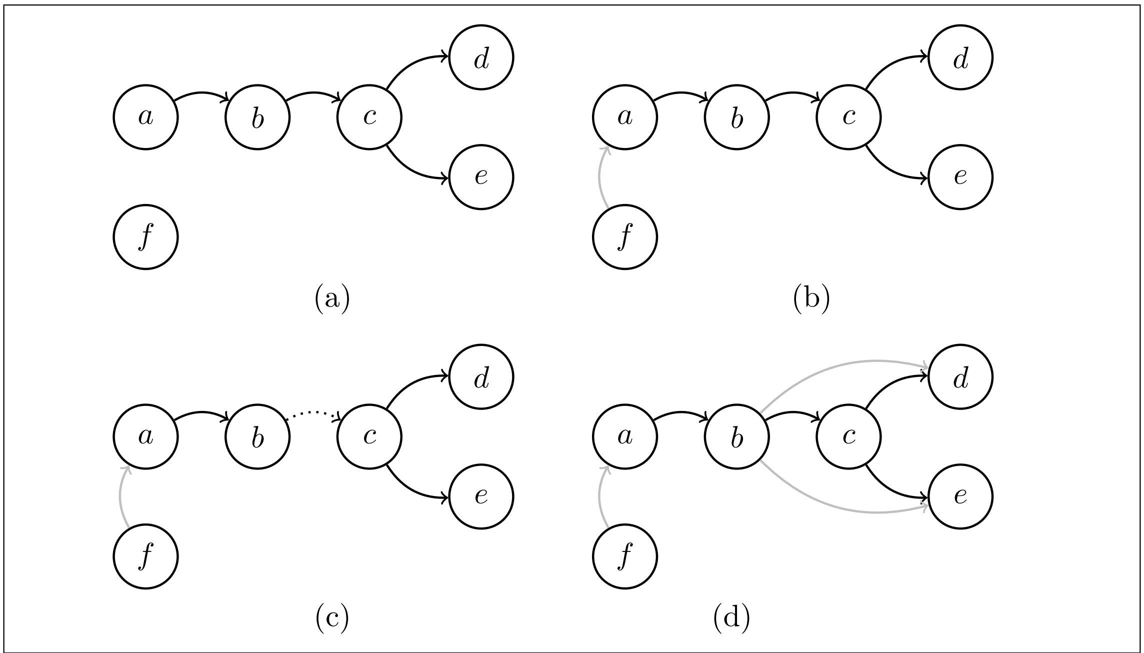


Figure 8: Example candidate AFs for Algorithm 1

After that it is verified that no modified AF with at most one modification (local update) achieves the strict enforcement under preferred semantics.

For two possible modifications, say the MaxSAT call returns an assignment corresponding to the AF in Figure 8(c). Due to the definition of the MaxSAT instance, we know that $\{b, f\}$ *is admissible in this candidate AF, which is like the previous one, except for removal of the attack from* b *to* c*. Here* $\{b, f\}$ *is again admissible, and the unique preferred extension is* $\{b, c, f\}$*, which is again verified not to be a solution. After checking all AFs that enforce* $\{b, f\}$ *to be admissible with at most one modification, the algorithm proceeds to at most three modifications, where a possible solution can be found, as illustrated in Figure 8(d).*

4 Related Notions to Enforcement

In this section, we overview several notions that are closely related to the enforcement setting described previously.

4.1 Update Using Logical Translations

YALLA (Yet Another Logic Language for Argumentation) [45] is a first-order logical language that allows to describe argumentation frameworks and their semantics. Then, operations related to enforcement can be defined through belief change theory, especially belief update [62].

Let us briefly describe the syntax and semantics of YALLA formulas. It is assumed that argumentation frameworks are built from a given *universe* $F_U = (A_U, R_U)$. This means that for any AF $F = (A, R)$, $A \subseteq A_U$ and $R \subseteq R_U \cap (A \times A)$. We write $k = |A_U|$ the number of arguments in the universe. A YALLA formula (or more precisely, YALLA_U) is a well-formed first order logic formula such that:

- the set of constant symbols is $V_{const} = \{c_\perp, c_1, \ldots, c_p\}$ where $p = 2^k - 1$;
- the set of function symbols is $V_{func} = \{\text{union}^2\}$;
- the set of predicate symbols is $V_{pred} = \{\text{on}^1, \rhd^2, \subseteq^2\}$.

The semantics of YALLA is defined through a structure associated with an AF $F = (A, R)$ built on the universe F_U. The domain of this structure is $D = 2^{A_U}$, and it is associated with an interpretation such that:

- the constant symbol $c_\perp$ is associated with the empty set; each constant symbol c_i $(i \in \{1, \ldots, 2^k - 1\})$ is associated with a different non-empty element of D;
- the union function symbol is associated with the binary set-theoretic union over D;
- the on predicate symbol is associated to the characterization function of subsets of A, i.e. $\text{on}(S)$ is true if and only if $S \subseteq A$;
- the predicate symbol $\rhd$ is associated with the set-attack relation induced by R, i.e. $S_1 \rhd S_2$ if and only if $S_1 \subseteq A$, $S_2 \subseteq A$, and $\exists a_1 \in S_1, a_2 \in S_2$ such that $(a_1, a_2) \in R$;
- the predicate symbol $\subseteq$ is associated with the classical inclusion relation over D.

Some axioms are added to the theory in order to guarantee the meaning of the YALLA formulas. For instance, if a set S_1 is included in A, then any subset of S_1 is included in A as well: this is formalized by $\forall x, y, (\text{on}(x) \wedge y \subseteq x) \Rightarrow \text{on}(y)$. A full description of the YALLA axioms is out of the scope of this chapter; we refer the interested reader to [45] for more details on this topic.

An argumentation framework $F = (A, R)$ can be described with the formula

$$\Phi_F = \text{on}(A) \wedge \bigwedge_{x \in A_U \setminus A} \neg\,\text{on}(\{x\}) \wedge \bigwedge_{(x,y) \in R} (\{x\} \triangleright \{y\}) \wedge \bigwedge_{(x,y) \in R_U \setminus R} \neg(\{x\} \triangleright \{y\})$$

Then, the principles underlying extension-based semantics can also be encoded as YALLA formulas. Given the structure associated with an AF $F = (A, R)$,

- the term t is conflict-free if the formula $\Phi_t^{cf} = \text{on}(t) \wedge \neg(\{t\} \triangleright \{t\})$ is valid;
- the term t_1 defends the term t_2, denoted by $t_1 \triangleright\!\triangleright t_2$, if the formula $\forall t_3, ((singl(t_3) \wedge t_3 \triangleright t_2) \rightarrow (t_1 \triangleright t_3))$ is valid, where $singl(t)$ is a formula that is valid if t is a singleton.

The combination of these formulas allows to characterize the admissible sets (i.e. the terms that satisfy of $\Phi_t^{ad} = \Phi_t^{cf} \wedge (t \triangleright\!\triangleright t))$. This is the basics of YALLA encoding for the classical Dung's semantics. Additional constraints in the formulas yield encodings Φ_t^{σ} for the other semantics.

Then, belief update rationality postulates and operators [62] are adapted to take into account the universe $F_U = (A_U, R_U)$. A set of authorized transitions (corresponding to what we call a modification type) is $T \subseteq \Gamma_U \times \Gamma_U$, where Γ_U is the set of all AFs built on the universe F_U. Then, roughly speaking, an update operator $\diamond_T$ is such that, if ϕ is a YALLA formula characterizing an AF F, then for any formula α, $\phi \diamond_T \alpha$ characterizes an AF G such that $(F, G) \in T$.[6]

Finally, enforcing an extension in an AF F can be achieved by updating the formula Φ_F:

$$\Phi_F \diamond_T \Phi_{c_i}^{\sigma}$$

characterizes the AFs that enforce S_i in F, under the modification type T and the semantics σ, where c_i is the YALLA constant symbol that corresponds to the set of arguments S_i, and Φ_t^{σ} is valid if and only if the term t corresponds to a σ-extension (similarly to the way Φ_t^{cf}, described previously, characterizes conflict-free sets).

Another logic-based approach is that of [48], which proposes to translate the argumentation framework and the semantics into logic, to perform the enforcement. In this case, the Dynamic Logic of Propositional Assignments (DL-PA) by [5], is used to represent update operators as executable programs. The piece of information which causes the update is a formula about acceptance statuses, which should be satisfied by at least one extension of the result (credulous enforcement of the formula) or by each extension of the result (sceptical enforcement of the formula).

[6]This is actually slightly more subtle than that, since YALLA formulas can characterize sets of AFs.

Forbus' update operator is used to change minimally the attack relation such that the extensions of the new argumentation framework comply with the expected enforcement. An extension of [48] is proposed by [50], which considers also addition and removal of arguments, and by applying the framework to an access control case. Then, [49] generalizes the previous two approaches.

Let us mention that these kinds of approaches based on a belief update operation allow richer forms of enforcement, since complex information about the sets of arguments and the attacks in the AFs can be described. Also, other kinds of belief change operations (e.g. belief revision [63] or belief contraction [36]) could be defined in these contexts. We refer the interested reader to [53] for more details on the relation between belief change and argumentation.

4.2 Status Enforcement

Status enforcement [72] is defined as an operator where two sets of arguments are provided as input, that must be respectively positively and negatively enforced. This operation does not fit the framework described previously, since it is supposed that there is only one set of arguments given in input, that must have exactly one acceptance status with respect to some reasoning mode (see Definition 3.1).

Formally, given an AF $F = (A, R)$, P and N two subsets of A such that $P \cap N = \emptyset$, and σ a semantics,

- the AF $G = (A, R')$ is a credulous status enforcement of (P, N) in F with respect to σ if $P \subseteq \bigcup \sigma(G)$ and $N \cap \bigcup \sigma(G) = \emptyset$;
- the AF $G = (A, R')$ is a sceptical status enforcement of (P, N) in F with respect to σ if $P \subseteq \bigcap \sigma(G)$ and $N \cap \bigcap \sigma(G) = \emptyset$.

In words, status enforcement consists in finding G such that every argument in P is credulously (respectively sceptically) accepted in G, and every argument in N is not credulously (respectively sceptically) accepted in G.

Complexity issues for optimal status enforcement, i.e. finding G such that $d(F, G) = |(R \setminus R') \cup (R' \setminus R)|$ is minimal, have been investigated by [72]. Similarly to complexity for optimal extension enforcement (Section 3.3), the complexity results concern a decision problem related to the optimization problem under consideration.

Credulous status enforcement
Instance: an AF $F = (A, R)$, $P \subseteq A$ and $N \subseteq A$ s.t. $P \cap N = \emptyset$, integer $k \geq 0$, and a semantics σ.
Question: Does there exist an AF $G = (A, R')$ such that $P \subseteq \bigcup \sigma(G)$ and $N \cap \bigcup \sigma(G) = \emptyset$ and $d(F, G) \leq k$?

semantics σ	$N = \emptyset$ credulous	$N = \emptyset$ sceptical	N unrestricted credulous	N unrestricted sceptical
cf	in P	trivial	in P	trivial
ad	NP-c	trivial	Σ_2^P-c	trivial
co	NP-c	NP-c	Σ_2^P-c	NP-c
gr	NP-c	NP-c	NP-c	NP-c
stb	NP-c	Σ_2^P-c	Σ_2^P-c	Σ_2^P-c
pr	NP-c	Σ_3^P-c	Σ_2^P-c	Σ_3^P-c

Table 5: Complexity of status enforcement

Sceptical status enforcement
Instance: an AF $F = (A, R)$, $P \subseteq A$ and $N \subseteq A$ s.t. $P \cap N = \emptyset$, integer $k \geq 0$, and a semantics σ.
Question: Does there exist an AF $G = (A, R')$ such that $P \subseteq \bigcap \sigma(G)$ and $N \cap \bigcap \sigma(G) = \emptyset$ and $d(F, G) \leq k$?

Two cases are considered: the general case, and the restricted case where $N = \emptyset$ (i.e. only positive arguments must be enforced). Table 5 presents the complexity of these problems for various semantics.

MaxSAT and CEGAR based algorithms in the same spirit as algorithms for extension enforcement (Section 3.3.2) are also provided.

4.3 Control Argumentation Frameworks

Now we introduce a concept that can be interpreted as a variant of enforcement under uncertain information. Control Argumentation Frameworks (CAFs) [46] are AFs where arguments and attacks are split in three distinct parts:

- the fixed part is made of arguments and attacks that are unquestionably in the system;
- the uncertain part is made of arguments and attacks that may belong to the system, as well as “undirected" attacks: in this case there is for sure a conflict between arguments, but the actual direction is uncertain;
- the control part is made of arguments and attacks that may be used by the agent.

The sets of fixed, uncertain and control arguments are disjoint, as well as the various sets of attacks. Roughly speaking, the fixed part corresponds to certain knowledge, i.e., elements that cannot be influenced neither by the agent nor by its environment (we use “environment" in a wide sense, it also includes other agents). On the contrary, the uncertain part models the agent’s knowledge (and beliefs) about the environment (and the other agents); in realistic scenarios, this knowledge is by nature uncertain. Finally, the control part corresponds to the agent’s possible actions. When the agent selects a subset of the control arguments and attacks (called a configuration), then it defines a *configured CAF*, that is the same CAF where the control arguments (and the associated attacks) that have not been selected have been removed. The uncertain part of the CAF induces a set of completions, i.e. classical AFs that are compatible with the knowledge encoded in the CAF. This notion is borrowed from Incomplete Argumentation Frameworks [42; 23; 25].

The notion of controllability of a CAF, with respect to a given target set of arguments, is directly related to enforcement. This target is defined as a subset of the fixed arguments, that is expected to belong to each (or some) extension of each completion. The agent needs to find a configuration that reaches this target. Let us exemplify these concepts.

Example 4.1. *Figure 9 describes a CAF, where the set of fixed arguments is* $\{f_1, f_2, f_3, f_4, f_5\}$*, the only uncertain argument is u (dashed square argument), and the control arguments are* $\{c_1, c_2, c_3\}$ *(bold square arguments). The plain arrows represent fixed attacks (e.g.* (f_2, f_1) *is fixed); the dotted arrow* (f_5, f_1) *means that it is uncertain whether* f_5 *actually attacks* f_1 *or not; the symmetric dashed arrow* (u, f_4) *means that there is for sure a conflict between u and* f_4*, but the actual direction is uncertain (it could be* (u, f_4)*, or* (f_4, u)*, or both at the same time). Finally, the bold arrows represent control attacks, they are related to the control arguments that can be selected by the agent.*

We suppose that the target $T = \{f_1\}$ *must belong to each stable extension. Without control arguments, this is not possible: there are, for instance, completions where* f_5 *attacks* f_1*, and in this case* f_1 *is not defended. However, with the configuration* $\{c_1, c_3\}$*,* f_1 *will be defended against every possible threat coming from the uncertain part:* c_1 *defends* f_1 *against* f_5*, and* c_3 *defends* f_1 *against u (that is an undirect threat, since u may defeat* f_4*, making then* f_2 *and* f_3 *acceptable). Similarly,* $\{c_2\}$ *is a valid configuration, since it allows to guarantee that* $\{f_1\}$ *is included in every stable extension of every completion.*

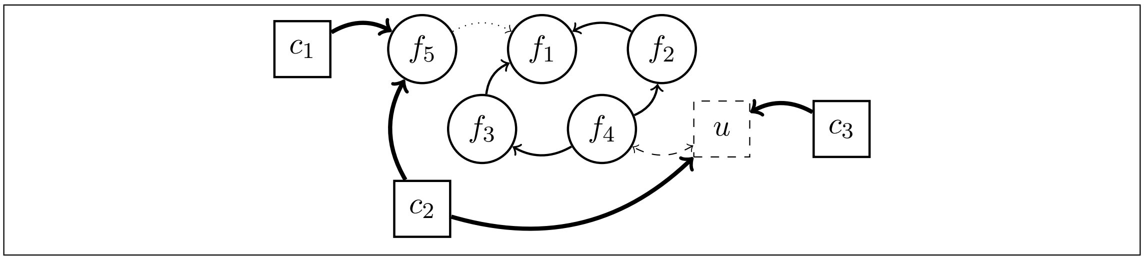


Figure 9: An example of control argumentation framework

Controlling a CAF can be seen as enforcing (non-strictly) an extension in presence of uncertainty. Intuitively, for an AF $F = (A, R)$ and a set of arguments E to be enforced through strong expansion, we can define a CAF that is controllable with respect to E if and only if it is possible to enforce E in F. Indeed, the arguments A and attacks R correspond to the fixed part of the CAF, while the uncertain part is empty. Then, for each $a \in A$, a control argument c_a with a control attack (c_a, a) is added. If E can be enforced in F, then the CAF is controllable (where the configuration to be chosen consists in the set of control arguments that do not attack E). On the opposite, if E cannot be enforced through a strong expansion, then the CAF is not controllable: indeed, the CAF configured by a control configuration is a strong expansion of F, thus E cannot be accepted in this configured CAF. We give a simple example of this transformation.

Example 4.2. *Let $F = (A, R)$ be the AF given at Figure 10a. We consider the grounded semantics: $gr(F) = \{\emptyset\}$. Let $E_1 = \{a\}$ be a set of arguments to be (non-strictly) enforced through a strong expansion. This enforcement is possible: for instance, the AF G that is a strong expansion of F where a new argument attacks b yields the expected result. Such an AF G corresponds to the CAF (given at Figure 10b) after it has been configured by $\{c_b\}$ (i.e. the argument c_a and the attack (c_a, a) are removed). So we observe that this CAF is controllable with respect to E_1 and the grounded semantics. On the opposite, $E_2 = \{a, b\}$ cannot be enforced in F with strong enforcement (since it is not conflict-free), and similarly there is no way to configure the CAF with respect to E_2 and the grounded semantics.*

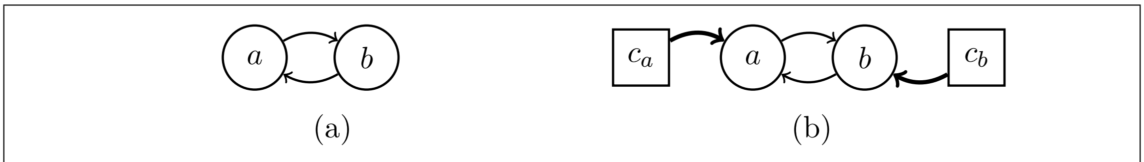


Figure 10: Transforming an AF into a CAF

In the previous example, we show how non-strict enforcement under strong expansion can seen as controlling a CAF. But more generally, since control arguments can attack each others, non-strict enforcement under a normal expansion can also be “translated" in controlling a CAF. On the opposite, configuring a CAF for controlling a target T can be interpreted as enforcing T in all the completions of the CAF with the same normal expansion (where the added arguments and attacks are chosen in the control part).

Let us also briefly mention that detailed complexity results and algorithms for reasoning with CAFs have been provided in [71], and [66] defines a weaker form of controllability, that relies on one completion instead of the whole set.

Applying CAFs to Automated Negotiation Let us briefly described how enforcement (or more precisely, CAFs) has been used in a context of automated negotiation [47]. The idea is to represent the (uncertain) knowledge of an agent about her opponent with a CAF. Indeed, negotiation has more chance to reach an agreement if agents have some knowledge about each other; however it is unrealistic to consider that opponent modelling can be done without incomplete or uncertain information. The theory of a negotiating agent is thus made of two parts: a classical AF that represents the agent's personal knowledge, and a CAF that represents her knowledge about her opponent.

It is supposed that agents negotiate about a set of (mutually exclusive) offers $\mathcal{O}$. Each offer may be supported, in AF_1 (the personal knowledge of agent 1), by 0, 1 or several *practical arguments*, i.e. arguments whose conclusions correspond to actions or decisions. The other arguments are *epistemic arguments*, they support knowledge and beliefs. The knowledge of agent 1 about agent 2 is represented in CAF_1^2. The fixed and uncertain parts are supposed to be built from the actual AF of agent 2: the assumption is made that there can be uncertainty (represented in the CAF), ignorance (some arguments or attacks of agent 2 may not appear in the CAF at all), but no mistake (there is no attacks or arguments that appear in CAF_1^2 but not in the personal AF of agent 2). Finally, the control part of CAF_1^2 is made of arguments and attacks chosen in AF_1, that are supposed to be used by agent 1 in order to make its target accepted. Similarly, AF_2 is the personal knowledge of agent 2, and CAF_2^1 represents the (uncertain) knowledge of agent 2 about agent 1.

Each agent selects its preferred offer $o \in \mathcal{O}$ according to its personal knowledge: o has to be supported by a practical argument that is accepted in AF_1; if several offers can be chosen, an assumption is made that the agent has a preference ranking over offers. When the preferred offer o of agent 1 is chosen, she uses her knowledge about agent 2 in order to persuade her to accept o: agent 1 searches for a practical argument in CAF_1^2 that supports o. If such an argument a exists, then three options are possible:

- if a is accepted in each completion of CAF_1^2 without using any control arguments, then agent 1 makes an offer to agent 2 (offer o, supported by argument a);

- otherwise, if a is accepted with the use of some control arguments $c_1, \ldots,$ c_k, then agent 1 can again make an offer (offer o, supported by argument a, that is accepted because of $c_1, \ldots, c_k$);

- in the last case, a is not accepted even with control arguments, then agent 1 searches for another argument that supports offer o in CAF_1^2.

In the first two cases, if agent 2 accepts the argument a (with, or without control arguments), then the negotiation is a success: offer o is accepted. Otherwise, agent 2 gives to agent 1 the reasons why she rejects a (for instance, she knows some arguments that agent 1 does not know). If agent 1 knows other arguments that support o in CAF_1^2, the process is repeated. Otherwise, this is the end of the round: agents switch their roles, and now agent 2 will choose her preferred offer o', and use her CAF in order to persuade agent 1 to accept o'.

The whole process goes on, until either the agents agree on some offer (in that case, the negotiation is a success), or they do not have available offers (the negotiation fails).

Let us illustrate the process, with an example borrowed from [47].

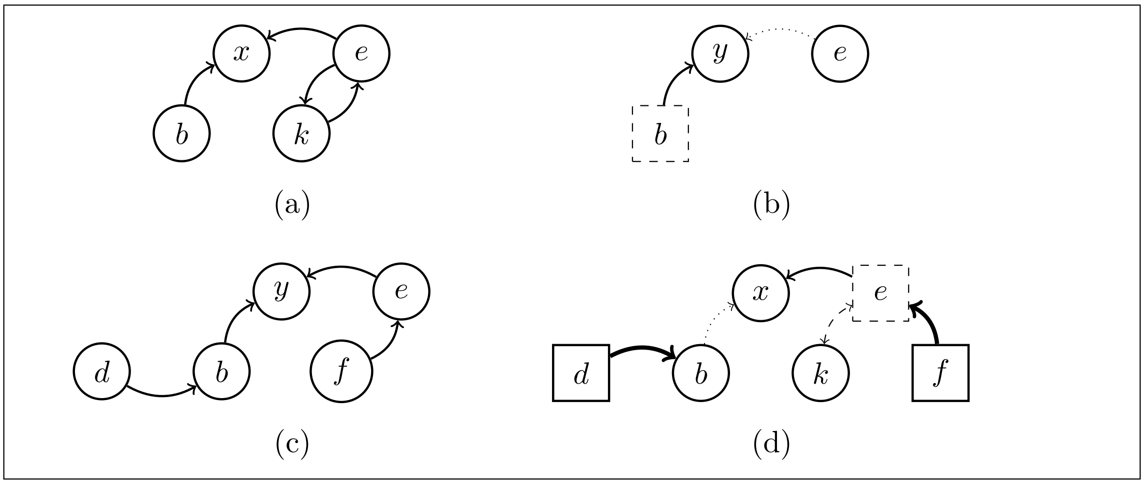


Figure 11: Initial theories of agents 1 and 2

Example 4.3. *Figure 11 describes the negotiation theories of two agents. More precisely, the AFs AF_1 and AF_2 (respectively Figure 11a and Figure 11c) correspond to the personal knowledge of (respectively) agent* 1 *and agent* 2*, while CAF_1^2 (Figure 11b) represents the (uncertain) knowledge of agent* 1 *about agent* 2*, and vice-versa for CAF_2^1 (Figure 11d). We suppose that both agents use the stable semantics for reasoning, and that there is one offer o, that is supported by arguments x and y. Before starting the negotiation, agent* 1 *has no reason to accept the offer o (since its supporting argument x is rejected in AF_1), while agent* 2 *accepts o since y is accepted in AF_2. If agent* 1 *starts the negotiation, she has no offer to propose (since there is no accepted argument in AF_1 that supports some offer), so the token has to go to agent* 2*.*

Agent 2 *can make an offer. The goal of agent* 2 *is to persuade agent* 1 *to accept the offer o, using arguments that agent* 1 *already knows. This means that she needs to make agent* 1 *modify her AF in order to accept x (since x is the only argument that supports the offer o in CAF_2^1). This persuasion phase goes first through a step that do not use the control part of the CAF: if x is accepted in the CAF with no control argument, agent* 2 *can send to agent* 1 *the message "offer o, supported by the accepted argument x". In the present example, this is not the case: there are completions where x is rejected (for instance, the ones where the attack (b, x) exists). So, in the next step, agent* 2 *searches for a control configuration that allows to make x accepted in each completion. Here, the configuration is the full set of control arguments $\{d, f\}$. Agent* 2 *can then send the message "offer o, supported by the argument x, that is accepted because d attacks b and f attacks e".*

Receiving this message triggers some updates in agent 1*'s knowledge. First, she can add the arguments d and f (as well as the attacks (d, b) and (f, e)) in CAF_1^2. Moreover, while argent b was initially uncertain in the CAF, it can now become a fixed argument: since agent* 2 *sends a message about the argument b, it certainly means that agent* 2 *knows this argument. Then, agent* 1 *can also add these arguments and attacks in her AF. The updated AF_1 and CAF_1^2 are shown at Figure 12. Since in the update AF_1, the argument x is accepted, agent* 1 *can stop the negotiation by accepting the offer o.*

Let us suppose that agent 1 *has, e.g., some argument i attacking d, then instead of accepting the offer o, she sends the message "reject the offer, because i attacks d". Then agent* 2 *updates her CAF, and the process continues as illustrated previously until reaching the negotiation success (if some offer can be accepted by both agents) or failure (if no offer can be accepted by both agents, even when exchanging arguments for defending them).*

Experiments [47] have shown that control arguments and attacks help to increase

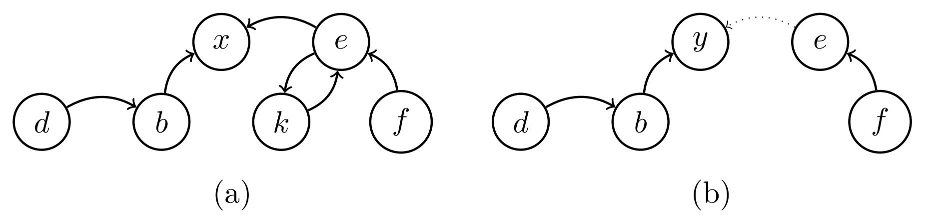


Figure 12: The updated theory of agent 1

the agreement rate, even when the percentage of uncertainty in CAFs is high.

4.4 Enforcement under Constraints

We have already seen several approaches to enforcement, and variants to enforcement. Common to many approaches are restrictions on the allowed modifications, such as allowing only expansions, local updates, other types of modifications, or defining allowed modifications via formulas, such as in YALLA.

More broadly, one can impose *constraints* on enforcements. Such constraints can have many different shapes or forms (expansions, deletions, specification as a formula, etc.). In general, constraints can be very useful for applications of enforcement operators. Going into a slightly different direction from before in this chapter, consider the following example.

Example 4.4. *Say we have knowledge about two arguments a and b, and we wish to enforce non-acceptability of a, e.g., because argument a counters a desirable argument. An expansion by c and attack (c, a) does the trick. However, say, in addition, that a is a sub argument of argument b, when inspecting the contents of the arguments. In such a case it seems adequate to require that c attacks the super argument of a, as well, i.e., we want also to add the attack (c, b).*

As suggested by the example, in some situations we may be faced with circumstances that may require specific expansions, or rather ruling out certain expansions. For instance, only considering those expansions that satisfy the condition that if an attack from some argument onto a is added, so must the same attacker also attack b.

Such conditions are not directly captured by the main types of modifications represented in this chapter, but can be incorporated into enforcement, as well. In [84], several families of constraints are considered, and the survey [53] discusses constraints of dynamics in argumentation, in general. We refer the reader for details to these papers, but highlight a particular type of constraint: implications of presence of arguments and attacks. By allowing constraints that take the form of implications, e.g., of the form mentioned above for attacks, one can specify that attacks on sub arguments must "propagate" to super arguments, which is present in many instances of formal approaches to structured argumentation [44; 67].

4.5 Other related works

Beside extension enforcement, adding or removing arguments or attacks to an AF can be seen as another form of enforcement, on the structure of the AF. This relates to dynamic aspects of argumentation. As already mentioned, [30; 38] are among the first approaches that studied the changes implied by such structural enforcements. Other similar approaches are detailed by [53].

In [82] the authors studied the question of how to *repair* an AF if nothing is credulously/sceptically accepted. More precisely, the main aim is to restore consistency via removing certain (minimal) sets of arguments or attacks. Note that enforcing a certain non-empty set can be seen as a special kind of repairing given that we are faced with no credulously accepted arguments. The notion of C-restricted semantics [20] is related to enforcement too. It can be shown that a set of arguments is a C-restricted extension if and only if it can be (non-strictly) enforced with a restricted form of expansion.

Normal expansions of AFs have been used for other purposes related to enforcement. For instance, [33] describes a framework where an agent's knowledge is represented by an AF F and a propositional formula ϕ that represents an integrity constraint about the complete labellings of the AF. The agent's knowledge is said to be inconsistent if none of the complete labellings satisfies the constraint. Two approaches are proposed for restoring consistency. The first one is a direct use of a normal expansion: the authors have proven that there exists a normal expansion of F that is consistent with ϕ (under some minimal assumption about the consistency of ϕ). The second approach also uses normal expansion, but only after a first step that consists in revising [63] the complete labellings of F by ϕ, in order to compute the so-called fallback beliefs of the agent. Then, a normal expansion allows to obtain a new AF that is consistent with the fallback beliefs. Contrary to the first approach (based only on an expansion), this one guarantees that the agent's complete labellings are as close as possible to the initial complete labellings.

Quite recently, the inverse problem to extension enforcement was studied, namely the problem of *extension removal* [18]. That is: given an AF F and a set of extensions $\mathcal{E}$, identify an AF H that is as close as possible to F but has none of the extensions in $\mathcal{E}$. In the same way as enforcement shifts revision to the level of extensions, extension removal shifts contraction to the level of extensions.

The approach by [56] aims at checking if a set E of sets of arguments can be the set of extensions of any argumentation framework F with respect to a given semantics σ. This property is named *realizability* of E with respect to σ. Realizability can be seen as a form of enforcement, where a set of extensions has to be enforced, and all the necessary structural changes on the argumentation framework (nothing is known about beforehand) can be done to achieve this.

A further related approach to enforcement is that of learning AFs or synthesis of AFs [79; 78; 75]. In brief, the aim is to construct an AF from certain information available. Different from deterministic logical approaches that construct an AF from a knowledge base (see structured argumentation approaches, e.g., in [6]), in AF learning or synthesis the information available might not uniquely determine an AF. In the AF synthesis problem [75], for instance, information about the semantics is given, and the task is to construct an AF that as best as possible matches the given semantic information. In this way, AF synthesis is related to realizability (see above), as well.

5 Conclusion

This chapter has offered an overview of the notion of enforcement in abstract, formal argumentation. A focus has been done on extension enforcement, on its general characterization, and on how it can be achieved: the various changes that can be applied to the structure of the argumentation framework, and/or to the semantics, considering that these changes should be minimal. Results about the complexity of enforcement, and algorithms, showing the feasibility of this approach, have also been presented.

If a general context and a number of specific approaches have been described, many additional proposals exist and keep on being proposed, showing the liveliness of the field. Applications of these formal approaches have also been outlined, and they should be developed in the future.

Regarding future work, several lines of research appear intriguing. Regarding formal foundations, we surveyed the state of the art, yet several directions are open, such as considering further argumentation semantics and their effect on possibility, impossibility, or (computational) cost of (optimal) enforcement. Moreover, different types of modifications can be considered as well, reflecting different updates on the given argumentation.

Beyond Dung's classical argumentation framework, the notion of enforcement can be defined and applied to any enriched argumentation framework, such as value-based argumentation frameworks (see Chapter 5 [4]), or frameworks with higher-order bipolar interactions (see Chapter 1 [37]), or with quantitative additions like probabilistic argumentation (see Chapter 7 [61]). The notion can also be extended to semantics other than extension-based, for instance their labelling-based counterparts [34], or ranking-based semantics [32].

Chapter 4 [24] studies (among other notions) Incomplete Argumentation Frameworks (IAFs), that are strongly related to CAFs described in this Chapter. Moreover, the possibility of enforcing a set of arguments can be intuitively associated with the notion of possible acceptance in IAFs.

Enforcement is also related to the notion of dialogue (see Chapter 9 [29]), where it can be put in practice, and to that of strategic argumentation (see Chapter 10 [59]). To go further, an empirical cognitive study of enforcement might be conducted, as it has been done for other argumentation notions (see Chapter 14 [39]).

Acknowledgments

This work was supported by the Austrian Science Fund (FWF, P30168-N31). Furthermore, this work was supported by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI Dresden/Leipzig".

References

[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[2] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215, 2002.

[3] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[4] Katie Atkinson and Trevor Bench-Capon. Value-based argumentation. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 5. College Publications, 2021.

[5] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Dynamic logic of propositional assignments: a well-behaved variant of pdl. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 143–152. IEEE, 2013.

[6] P. Baroni, M. Giacomin, and B. Liao. Locality and modularity in abstract argumentation. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der

Torre, editors, *Handbook of Formal Argumentation*, chapter 19. College Publications, February 2018.

[7] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. Abstract argumentation frameworks and their semantics. In *Handbook of Formal Argumentation*, chapter 4. College Publications, February 2018.

[8] Pietro Baroni, Federico Cerutti, Paul E. Dunne, and Massimiliano Giacomin. Automata for infinite argumentation structures. *Artificial Intelligence*, 203:104–150, 2013.

[9] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.

[10] R. Baumann. On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 14. College Publications, February 2018. also appears in IfCoLog Journal of Logics and their Applications 4(8):2779-2886.

[11] Ringo Baumann. Splitting an argumentation framework. In James P. Delgrande and Wolfgang Faber, editors, *Proc. LPNMR*, volume 6645 of *Lecture Notes in Computer Science*, pages 40–53. Springer, 2011.

[12] Ringo Baumann. Normal and strong expansion equivalence for argumentation frameworks. *Artificial Intelligence*, 193:18–44, 2012.

[13] Ringo Baumann. What does it take to enforce an argument? minimal change in abstract argumentation. In Luc De Raedt, Christian Bessiere, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, editors, *Proc. ECAI*, volume 242, pages 127–132. IOS Press, 2012.

[14] Ringo Baumann. Context-free and context-sensitive kernels: Update and deletion equivalence in abstract argumentation. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proc. ECAI*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 63–68. IOS Press, 2014.

[15] Ringo Baumann. *Metalogical Contributions to the Nonmonotonic Theory of Abstract Argumentation.* College Publications, 2014.

[16] Ringo Baumann. *On the Existence of Characterization Logics and Fundamental Properties of Argumentation Semantics.* habilitation treatise, Leipzig University, November 2019.

[17] Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari, editors, *Proc. COMMA*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 75–86. IOS Press, 2010.

[18] Ringo Baumann and Gerhard Brewka. Extension removal in abstract argumentation - an axiomatic approach. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 2670–2677, 2019.

[19] Ringo Baumann, Gerhard Brewka, and Renata Wong. Splitting argumentation frameworks: An empirical evaluation. In Sanjay Modgil, Nir Oren, and Francesca Toni, editors, *Proc. TAFA, Revised Selected Papers*, volume 7132 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2011.

[20] Ringo Baumann, Wolfgang Dvorák, Thomas Linsbichler, and Stefan Woltran. A general notion of equivalence for abstract argumentation. *Artif. Intell.*, 275:379–410, 2019.

[21] Ringo Baumann and Christof Spanring. Infinite argumentation frameworks - On the existence and uniqueness of extensions. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, pages 281–295, 2015.

[22] Ringo Baumann and Christof Spanring. A study of unrestricted abstract argumentation frameworks. In Carles Sierra, editor, *Proc. IJCAI*, pages 807–813. ijcai.org, 2017.

[23] Dorothea Baumeister, Daniel Neugebauer, and Jörg Rothe. Credulous and skeptical acceptance in incomplete argumentation frameworks. In Sanjay Modgil, Katarzyna Budzynska, and John Lawrence, editors, *Proc. COMMA*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 181–192. IOS Press, 2018.

[24] Dorothea Baumeister, Daniel Neugebauer, and Jörg Rothe. Collective acceptability in abstract argumentation. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 4. College Publications, 2021.

[25] Dorothea Baumeister, Daniel Neugebauer, Jörg Rothe, and Hilmar Schadrack. Verification in incomplete argumentation frameworks. *Artificial Intelligence*, 264:1–26, 2018.

[26] Philippe Besnard and Sylvie Doutre. Checking the acceptability of a set of arguments. In *Proc. NMR*, pages 59–64, 2004.

[27] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[28] Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in argumentation systems: Exploring the interest of removing an argument. In Salem Benferhat and John Grant, editors, *Proc. SUM*, volume 6929 of *Lecture Notes in Computer Science*, pages 275–288. Springer, 2011.

[29] Elizabeth Black, Nicolas Maudet, and Simon Parsons. Argumentation-based dialogue. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 9. College Publications, 2021.

[30] Guido Boella, Souhila Kaci, and Leendert W. N. van der Torre. Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In Claudio Sossai and Gaetano Chemello, editors, *Proc. ECSQARU*, volume 5590 of *Lecture Notes in Computer Science*, pages 107–118. Springer, 2009.

[31] Guido Boella, Souhila Kaci, and Leendert W. N. van der Torre. Dynamics in argumentation with single extensions: Attack refinement and the grounded extension (extended version). In Peter McBurney, Iyad Rahwan, Simon Parsons, and Nicolas Maudet, editors, *ArgMAS Revised Selected and Invited Papers*, volume 6057 of *Lecture Notes in*

Computer Science, pages 150–159. Springer, 2009.

[32] Elise Bonzon, Jerome Delobelle, Sebastien Konieczny, and Nicolas Maudet. A comparative study of ranking-based semantics for abstract argumentation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, 2016.

[33] R. Booth, S. Kaci, T. Rienstra, and L. W. N. van der Torre. A logical theory about dynamics in abstract argumentation. In Weiru Liu, V. S. Subrahmanian, and Jef Wijsen, editors, *Proc. SUM*, volume 8078 of *Lecture Notes in Computer Science*, pages 148–161. Springer, 2013.

[34] Martin Caminada. On the issue of reinstatement in argumentation. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15, 2006, Proceedings*, volume 4160 of *Lecture Notes in Computer Science*, pages 111–123. Springer, 2006.

[35] Martin Caminada and Yining Wu. On the limitations of abstract argumentation. In *Benelux Conference on Artificial Intelligence*, 2011.

[36] Thomas Caridroit, Sébastien Konieczny, and Pierre Marquis. Contraction in propositional logic. *International Journal of Approximate Reasoning*, 80:428–442, 2017.

[37] Claudette Cayrol, Andrea Cohen, and Marie-Christine Lagasquie-Schiex. Higher-order interactions (bipolar or not) in abstract argumentation: A state of the art. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 1. College Publications, 2021.

[38] Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in abstract argumentation frameworks: Adding an argument. *Journal of Artificial Intelligence Research*, 38:49–84, 2010.

[39] Federico Cerutti, Marcos Cramer, Mathieu Guillaume, Emmanuel Hadoux, Anthony Hunter, and Sylwia Polberg. Empirical cognitive studies about formal argumentation. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 14. College Publications, 2021.

[40] E. M. Clarke, A. Gupta, and O. Strichman. SAT-based counterexample-guided abstraction refinement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(7):1113–1123, 2004.

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

[42] Sylvie Coste-Marquis, Caroline Devred, Sébastien Konieczny, Marie-Christine Lagasquie-Schiex, and Pierre Marquis. On the merging of dung's argumentation systems. *Artificial Intelligence*, 171(10-15):730–753, 2007.

[43] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. Extension enforcement in abstract argumentation as an optimization problem. In Qiang Yang and Michael J. Wooldridge, editors, *Proc. IJCAI*, pages 2876–2882. AAAI Press, 2015.

[44] Kristijonas Cyras, Xiuyi Fan, Claudia Schulz, and Francesca Toni. Assumption-based argumentation: Disputes, explanations, preferences. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 7, pages 365–408. College Publications, 2018.

[45] Florence Dupin de Saint-Cyr, Pierre Bisquert, Claudette Cayrol, and Marie-Christine Lagasquie-Schiex. Argumentation update in YALLA (yet another logic language for argumentation). *International Journal of Approximate Reasoning*, 75:57–92, 2016.

[46] Yannis Dimopoulos, Jean-Guy Mailly, and Pavlos Moraitis. Control argumentation frameworks. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proc. AAAI*, pages 4678–4685. AAAI Press, 2018.

[47] Yannis Dimopoulos, Jean-Guy Mailly, and Pavlos Moraitis. Argumentation-based negotiation with incomplete opponent profiles. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proc. AAMAS*, pages 1252–1260. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[48] S. Doutre, A. Herzig, and L. Perrussel. A dynamic logic framework for abstract argumentation. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proc. KR*, pages 62–71. AAAI Press, 2014.

[49] Sylvie Doutre, Andreas Herzig, and Laurent Perrussel. Abstract argumentation in dynamic logic: Representation, reasoning and change. In *Dynamics, Uncertainty and Reasoning*, pages 153–185. Springer, 2019.

[50] Sylvie Doutre, Faustine Maffre, and Peter McBurney. A Dynamic Logic Framework for Abstract Argumentation: Adding and Removing Arguments (regular paper). In Salem Benferhat, Karim Tabia, and Moonis Ali, editors, *Advances in Artificial Intelligence: From Theory to Practice - International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2017)*, volume 10351 of *Lecture Notes in Computer Science*, pages 295–305. Springer-Verlag, 2017.

[51] Sylvie Doutre and Jean-Guy Mailly. Comparison criteria for argumentation semantics. In Francesco Belardinelli and Estefania Argente, editors, *Proc. EUMAS/AT, Revised Selected Papers*, volume 10767 of *Lecture Notes in Computer Science*, pages 219–234. Springer, 2017.

[52] Sylvie Doutre and Jean-Guy Mailly. Semantic change and extension enforcement in abstract argumentation. In Serafín Moral, Olivier Pivert, Daniel Sánchez, and Nicolás Marín, editors, *Proc. SUM*, volume 10564 of *Lecture Notes in Computer Science*, pages 194–207. Springer, 2017.

[53] Sylvie Doutre and Jean-Guy Mailly. Constraints and changes: A survey of abstract argumentation dynamics. *Argument & Computation*, 9(3):223–248, 2018.

[54] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[55] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.

[56] Paul E Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Char-

acteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence*, 228:153–178, 2015.

[57] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 13, pages 631–688. College Publications, 2018.

[58] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proc. ICLP/SLP*, pages 1070–1080. MIT Press, 1988.

[59] Guido Governatori, Michael J. Maher, and Francesco Olivieri. Strategic argumentation. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 10. College Publications, 2021.

[60] Adrian Haret, Johannes Peter Wallner, and Stefan Woltran. Two sides of the same coin: Belief revision and enforcing arguments. In Jérôme Lang, editor, *Proc. IJCAI*, pages 1854–1860. ijcai.org, 2018.

[61] Anthony Hunter, Sylwia Polberg, Nico Potyka, Tjitze Rienstra, and Matthias Thimm. Probabilistic argumentation: A survey. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 7. College Publications, 2021.

[62] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In James F. Allen, Richard Fikes, and Erik Sandewall, editors, *Proc. KR*, pages 387–394. Morgan Kaufmann, 1991.

[63] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1992.

[64] Mark W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[65] Bei Shui Liao, Li Jin, and Robert C. Koons. Dynamics of argumentation systems: A division-based method. *Artificial Intelligence*, 175(11):1790–1814, 2011.

[66] Jean-Guy Mailly. Possible controllability of control argumentation frameworks. In *Computational Models of Argument - Proceedings of COMMA 2020, Perugia, Italy, September 4-11, 2020*, pages 283–294, 2020.

[67] Sanjay Modgil and Henry Prakken. Abstract rule-based argumentation. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 6, pages 287–364. College Publications, 2018.

[68] António Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and João Marques-Silva. Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints*, 18(4):478–534, 2013.

[69] Ilkka Niemelä. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence*, 25(3-4):241–273, 1999.

[70] Andreas Niskanen and Matti Järvisalo. Strong refinements for hard problems in argumentation dynamics. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*, pages 841–848, 2020.

[71] Andreas Niskanen, Daniel Neugebauer, and Matti Järvisalo. Controllability of control argumentation frameworks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1855–1861, 2020.

[72] Andreas Niskanen, Johannes P. Wallner, and Matti Järvisalo. Optimal status enforcement in abstract argumentation. In Subbarao Kambhampati, editor, *Proc. IJCAI*, pages 1216–1222. IJCAI/AAAI Press, 2016.

[73] Andreas Niskanen, Johannes P. Wallner, and Matti Järvisalo. Pakota: A system for enforcement in abstract argumentation. In Loizos Michael and Antonis C. Kakas, editors, *Proc. JELIA*, volume 10021 of *Lecture Notes in Computer Science*, pages 385–400. Springer, 2016.

[74] Andreas Niskanen, Johannes Peter Wallner, and Matti Järvisalo. Extension enforcement under grounded semantics in abstract argumentation. In Michael Thielscher, Francesca Toni, and Frank Wolter, editors, *Proc. KR*, pages 178–183. AAAI Press, 2018.

[75] Andreas Niskanen, Johannes Peter Wallner, and Matti Järvisalo. Synthesizing argumentation frameworks from examples. *Journal of Artificial Intelligence Research*, 66:503–554, 2019.

[76] Emilia Oikarinen and Stefan Woltran. Characterizing Strong Equivalence for Argumentation Frameworks. *Artificial Intelligence*, 175(14-15):1985–2009, 2011.

[77] Christos H. Papadimitriou. *Computational complexity.* Academic Internet Publ., 2007.

[78] Régis Riveret. On learning abstract argumentation graphs from bivalent statement labellings. In *Proc. ICTAI*, pages 190–195. IEEE Computer Society, 2016.

[79] Régis Riveret and Guido Governatori. On learning attacks in probabilistic abstract argumentation. In Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls, editors, *Proc. AAMAS*, pages 653–661. ACM, 2016.

[80] Gerard Sierksma and Yori Zwols. *Linear and integer optimization: theory and practice.* CRC Press, 2015.

[81] Christof Spanring. Hunt for the collapse of semantics in infinite abstract argumentation frameworks. In Claudia Schulz and Daniel Liew, editors, *Proceedings of the 2015 Imperial College Computing Student Workshop, ICCSW 2015*, volume 49 of *OASICS*, pages 70–77. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[82] Markus Ulbricht and Ringo Baumann. If nothing is accepted - repairing argumentation frameworks. *J. Artif. Intell. Res.*, 66:1099–1145, 2019.

[83] Leon van der Torre and Srdjan Vesic. The principle-based approach to abstract argumentation semantics. *The IfCoLog Journal of Logics and their Applications*, 4(8):2727–2780, 2017.

[84] Johannes P. Wallner. Structural constraints for dynamic operators in abstract argumentation. *Argument & Computation*, 11(1–2):151–190, 2020.

[85] Johannes P. Wallner, Andreas Niskanen, and Matti Järvisalo. Complexity results and algorithms for extension enforcement in abstract argumentation. *Journal of Artificial Intelligence Research*, 60:1–40, 2017.

Received May 2021