Journal of Artificial Intelligence Research 60 (2017) 1-40 Submitted 12/16; published 09/17

Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation

Johannes P. Wallner WALLNER@DBAI.TUWIEN.AC.AT
Institute of Information Systems,
TU Wien, Austria

Andreas Niskanen ANDREAS.NISKANEN@HELSINKI.FI
Matti Järvisalo MATTI.JARVISALO@HELSINKI.FI
Helsinki Institute for Information Technology HIIT,
Department of Computer Science,
University of Helsinki, Finland

Abstract

Argumentation is an active area of modern artificial intelligence (AI) research, with connections to a range of fields, from computational complexity theory and knowledge representation and reasoning to philosophy and social sciences, as well as application-oriented work in domains such as legal reasoning, multi-agent systems, and decision support. Argumentation frameworks (AFs) of abstract argumentation have become the graph-based formal model of choice for many approaches to argumentation in AI, with semantics defining sets of jointly acceptable arguments, i.e., extensions. Understanding the dynamics of AFs has been recently recognized as an important topic in the study of argumentation in AI. In this work, we focus on the so-called extension enforcement problem in abstract argumentation as a recently proposed form of argumentation dynamics. We provide a nearly complete computational complexity map of argument-fixed extension enforcement under various major AF semantics, with results ranging from polynomial-time algorithms to completeness for the second level of the polynomial hierarchy. Complementing the complexity results, we propose algorithms for NP-hard extension enforcement based on constraint optimization under the maximum satisfiability (MaxSAT) paradigm. Going beyond NP, we propose novel MaxSAT-based counterexample-guided abstraction refinement procedures for the second-level complete problems and present empirical results on a prototype system constituting the first approach to extension enforcement in its generality.

1. Introduction

Argumentation is a core topic in Artificial Intelligence (AI) (Bench-Capon & Dunne, 2007), with applications in e.g. decision support (Amgoud & Prade, 2009), legal reasoning (Bench-Capon, Prakken, & Sartor, 2009), and multi-agent systems (McBurney, Parsons, & Rahwan, 2012). Argumentation frameworks (AFs) proposed by Dung (1995) in the context of abstract argumentation provide the fundamental formal model for many approaches to argumentation in AI. Syntactically, AFs are directed graphs, where arguments are abstract entities represented by vertices. Conflicts among arguments are formalized as attacks, and represented with directed edges between arguments. Semantics of AFs—several of which have been proposed—specify criteria for arguments' acceptance, resulting in sets of jointly acceptable arguments called *extensions*.

Argumentation is inherently a dynamic process. Recently, several works have focused on fundamental aspects of argumentation dynamics in terms of AFs, in particular, expansions (Baumann, 2012a; Cayrol, de Saint-Cyr, & Lagasquie-Schiex, 2010), revision (Baumann & Brewka, 2015; Coste-Marquis, Konieczny, Mailly, & Marquis, 2014a, 2014b; Diller, Haret, Linsbichler, Rümmele, & Woltran, 2015; Booth, Kaci, Rienstra, & van der Torre, 2013), aggregation (Delobelle, Konieczny, & Vesic, 2015; Delobelle, Haret, Konieczny, Mailly, Rossit, & Woltran, 2016; Endriss & Grandi, 2017), and logical frameworks (Doutre, Herzig, & Perrussel, 2014). In this work, we focus on *extension enforcement* (Baumann, 2012b; Bisquert, Cayrol, de Saint-Cyr, & Lagasquie-Schiex, 2013; de Saint-Cyr, Bisquert, Cayrol, & Lagasquie-Schiex, 2016; Coste-Marquis, Konieczny, Mailly, & Marquis, 2015; Coste-Marquis et al., 2014b; Doutre et al., 2014), a specific form of AF dynamics with connections to belief revision, concerned with finding changes to a given AF in order to support a desired point of view, represented as a set of arguments, under pre-specified semantics.

The goal of extension enforcement is to modify a given argumentation framework in light of new information in a way that a given set becomes an extension (Baumann & Brewka, 2010). The extension enforcement problem in abstract argumentation was proposed by Baumann and Brewka (2010), where the (im)possibilities of enforcing a set of arguments were studied under different conditions, namely, addition of new arguments and attacks from and to them (i.e., under the so-called normal, strong and weak expansions). This approach is viable in e.g. dialogue-based settings, where the question is how to make (enforce) a certain point of view (extension) acceptable by drawing new information into the discussion (de Saint-Cyr et al., 2016). Baumann (2012b) generalized the problem to enforcing a set of arguments under *minimal change*[1], that is, the question is not only whether the set *can* be enforced (a decision problem), but *how many changes* to the original framework are required at minimum in order to achieve the desired enforcement (an optimization problem). The removal of arguments and attacks associated with them was studied by Bisquert et al. (2013), de Saint-Cyr et al. (2016), who also generalized the extension enforcement problem to arbitrary goals expressible in first-order logic. All of these approaches make the assumption that the initial attacks of the AF are fixed, and instead new arguments can arrive or old ones can leave.

However, as noted by Baumann and Brewka (2010) and Coste-Marquis et al. (2015), enforcement under these expansions is impossible in the general case. Motivated by this, the *argument-fixed extension enforcement problem* was proposed by Coste-Marquis et al. (2015), where instead of adding new arguments and attacks, the arguments are fixed, and the original attack structure may be subject to any change, viewed as an optimization problem by minimizing the number of changes necessary. This view guarantees that both the so-called *strict* and *non-strict* variants—the tasks being enforcing a given set of arguments exactly as an extension (strict enforcement) or ensuring that a given set of arguments is contained in an extension (non-strict enforcement)—of this argument-fixed extension enforcement problem always have a solution. A further motivation for the argument-fixed extension enforcement problem is in the context of uncertain attacks, which are amenable to change. Such uncertainty can occur, e.g., when an AF was generated from arguments based on enthymemes (Mailly, 2016), i.e., arguments and attacks are created from partially unspecified premises that are assumed—in a potentially erroneous manner—from common knowledge. Enthymemes (incomplete syllogisms) are often used in natural language for technical and strategical reasons, since in human communication one may assume a "common knowledge base" from which to draw unspoken, or implicit, assumptions. Yet another motivation, from a more al-

1. Note that Baumann (2012b) uses the word 'minimal' as a synonym for 'minimum'.

gorithmic perspective, for the argument-fixed extension enforcement problem is that it subsumes, via addition of further constraints, enforcement under expansions when the set of new arguments is bounded (Coste-Marquis et al., 2015).

In this work, we take the work of Coste-Marquis et al. (2015) considerably further in terms of complexity analysis of and system implementations for the argument-fixed extension enforcement problem. Indeed, while the complexity landscape of non-dynamic problems on AFs, including the credulous and skeptical reasoning tasks for a given fixed AF, is already well-established (Dunne & Wooldridge, 2009), the complexity of extension enforcement under different semantics and problem variants has not been thoroughly studied until now. In terms of complexity studies, to the best of our knowledge the most closely related recent results consider the complexity of problems in the contexts of strategic argumentation (Maher, 2016c, 2016a, 2016b) and incomplete knowledge in argumentation (Baumeister, Neugebauer, & Rothe, 2015a; Baumeister, Rothe, & Schadrack, 2015b). Furthermore, while several efficient systems for the NP-hard variants of non-dynamic problems are available (Cerutti, Dunne, Giacomin, & Vallati, 2014a; Cerutti, Giacomin, & Vallati, 2014b; Dvořák, Järvisalo, Wallner, & Woltran, 2014; Egly, Gaggl, & Woltran, 2010; Nofal, Atkinson, & Dunne, 2014), to our best knowledge the single existing system for extension enforcement was only recently proposed (Coste-Marquis et al., 2015) and supports extension enforcement only w.r.t. a specific AF semantics (the *stable* semantics). This article aims at bridging these gaps.

1.1 Contributions

Our main contributions are the following.

- We provide a nearly complete computational complexity map of argument-fixed extension enforcement, where the task is to enforce a given extension by modifying the attack relation of a given AF. Our results cover nine standard AF semantics and both the strict and non-strict variants of extension enforcement. For instance, we provide polynomial-time algorithms for strict enforcement under the admissible and stable semantics (the latter of which was in fact proposed to be solved using the NP-machinery of integer programming (IP) in Coste-Marquis et al., 2015); we show that most non-strict enforcement problems are NP-complete, along with strict enforcement under the complete and grounded semantics; and establish second-level completeness for strict enforcement under preferred and semi-stable semantics as well as for non-strict semi-stable and stage semantics.

- We propose algorithms for the NP-hard variants of the enforcement problems based on applying constraint optimization solvers. We detail maximum satisfiability (MaxSAT) encodings for the NP-complete problem variants, and, perhaps most interestingly, propose novel counterexample-guided abstraction refinement (CEGAR) (Clarke, Grumberg, Jha, Lu, & Veith, 2003; Clarke, Gupta, & Strichman, 2004) procedures for the second-level Σ_2^P-complete variants using optimization solvers as functional NP oracles. This results in a first extension enforcement system in its generality, allowing for optimally solving instances of both the non-strict and strict variants of NP-hard extension enforcement under various central AF semantics. We also provide an extensive empirical evaluation of our system implementation.

While our main focus is argument-fixed extension enforcement, we also shortly discuss the *normal*, *strong*, and *weak* variants (Baumann, 2012b) of enforcement, and explain how our declarative approach can be extended to capture some of such further variants of extension enforcement.

This article considerably extends a preliminary version of the work presented at AAAI-16, 30th AAAI Conference on Artificial Intelligence (Wallner, Niskanen, & Järvisalo, 2016), including full previously unpublished formal proofs, new empirical results on the scalability of the approach, as well as extended examples and discussion.

1.2 Paper Structure

The rest of this article is organized as follows. We start with necessary preliminaries on abstract argumentation and argumentation frameworks, maximum satisfiability, and the standard complexity classes central to the following discussion (Section 2), and continue by formally defining extension enforcement as studied in this work as an optimization problem (Section 3). We then proceed to the main contributions of this work: a thorough complexity analysis of strict and non-strict extension enforcement under various central semantics (Section 4), algorithms for NP-hard extension enforcement, including both direct MaxSAT encodings of the NP-complete variants of extension enforcement (Section 5) and counterexample-guided abstraction refinement algorithms for the problem variants that are complete for the second level of the polynomial hierarchy (Section 6), followed by a large-scale empirical evaluation of the algorithms (Section 7). Finally, before conclusions, we briefly discuss other variants of extension enforcement, to what extent the algorithms proposed in this work can be easily adapted to cover the further variants, as well as computational complexity of some of the further variants (Section 8).

2. Preliminaries

We start by recalling the necessary background for the rest of our discussion: argumentation frameworks in the context of abstract argumentation, the central argumentation semantics considered in this work, and background on maximum satisfiability and some of the standard complexity classes.

2.1 Argumentation Frameworks

We recall concepts related to argumentation frameworks (Dung, 1995) and their semantics (Baroni, Caminada, & Giacomin, 2011).

Definition 1. *An* argumentation framework (AF) *is a pair $F = (A, R)$, where A is a finite set of arguments and $R \subseteq A \times A$ is the attack relation. The pair $(a, b) \in R$ means that a attacks b. An argument $a \in A$ is* defended *(in F) by a set $S \subseteq A$ if, for each $b \in A$ such that $(b, a) \in R$, there exists a $c \in S$ such that $(c, b) \in R$.*

Example 1. *Let $F = (A, R)$ be an AF with $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (b, c), (c, d), (d, c), (d, e), (e, e)\}$. The corresponding graph representation is shown in Figure 1.*

Semantics for argumentation frameworks are defined through a function σ which assigns to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions. We consider for σ the functions $naive$, stb,

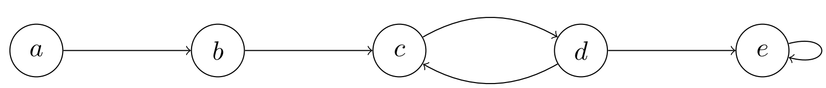


Figure 1: Example argumentation framework in Example 1.

adm, com, grd, prf, sem, and stg which stand for naive, stable, admissible, complete, grounded, preferred, semi-stable, and stage extensions, respectively. Towards the definition of these semantics we introduce some auxiliary formal concepts.

Definition 2. *Given an AF $F = (A, R)$, the* characteristic function $\mathcal{F}_F : 2^A \to 2^A$ *of F is $\mathcal{F}_F(S) = \{x \in A \mid x$ is defended by $S\}$. Moreover, for a set $S \subseteq A$, the* range of S *is $S_R^+ = S \cup \{x \mid (y, x) \in R, y \in S\}$.*

The semantics considered in this paper are as follows.

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

- *$S \in naive(F)$ iff there is no $T \in cf(F)$ with $S \subset T$;*
- *$S \in stb(F)$ iff $S_R^+ = A$;*
- *$S \in adm(F)$ iff $S \subseteq \mathcal{F}_F(S)$;*
- *$S \in com(F)$ iff $S = \mathcal{F}_F(S)$;*
- *$S \in grd(F)$ iff S is the least fixed-point of $\mathcal{F}_F$;*
- *$S \in prf(F)$ iff $S \in adm(F)$ and there is no $T \in adm(F)$ with $S \subset T$;*
- *$S \in sem(F)$ iff $S \in adm(F)$ and there is no $T \in adm(F)$ with $S_R^+ \subset T_R^+$;*
- *$S \in stg(F)$ iff there is no $T \in cf(F)$ with $S_R^+ \subset T_R^+$.*

It is a well-known fact that for any AF F it holds that $cf(F) \supseteq adm(F) \supseteq com(F) \supseteq prf(F) \supseteq sem(F) \supseteq stb(F)$. We use the term σ-extension to refer to an extension under a semantics $\sigma \in \{naive, stb, adm, com, grd, prf, sem, stg\}$.

Example 2. *Consider the AF F in Example 1 on page 4. For the semantics σ considered, the σ-extensions of F are enumerated in the following table.*

σ	$\sigma(F)$
$naive$	$\{\{a, c\}, \{a, d\}, \{b, d\}\}$
stb	$\{\{a, d\}\}$
adm	$\{\emptyset, \{a\}, \{a, c\}, \{a, d\}, \{d\}\}$
com	$\{\{a\}, \{a, c\}, \{a, d\}\}$
grd	$\{\{a\}\}$
prf	$\{\{a, c\}, \{a, d\}\}$
sem	$\{\{a, d\}\}$
stg	$\{\{a, d\}\}$

2.2 Central Complexity Classes

We briefly recall the complexity classes used in this paper (for more details we refer the reader to Papadimitriou, 1994). The complexity class P contains all problems that can be solved with a deterministic polynomial-time algorithm. The class NP contains all problems solvable via a non-deterministic polynomial-time algorithm. The class coNP is the class of complements of NP problems. Finally, we make use of the class $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^P$, which is the class of problems that can be decided with a non-deterministic polynomial-time algorithm that has access to an NP-oracle, i.e., can decide a problem in NP in constant time. A problem P is $\mathcal{C}$-hard if every problem P' in $\mathcal{C}$ is polynomial-time reducible to P, i.e., there is a polynomial-time algorithm that transforms an instance I' of P' to an instance I of P such that I' is a "yes" instance of P' if and only if I is a "yes" instance of P. A problem P is complete for a complexity class $\mathcal{C}$ if P is in $\mathcal{C}$ and P is $\mathcal{C}$-hard.

2.3 Maximum Satisfiability

In this work we present declarative encodings that can be used for solving extension enforcement optimally under various semantics. We employ maximum satisfiability (MaxSAT) as a well-suited declarative language. Specifically, we will present MaxSAT encodings as well as develop counterexample-guided abstraction refinement algorithms for solving the Σ_2^P-complete problem variants by applying the NP-encodings. For necessary background, we now recall the MaxSAT problem.

For a variable x, there are two literals, x and $\neg x$. A clause is a disjunction ($\vee$) of literals. A truth assignment is a function from variables to $\{0,1\}$. A clause c is satisfied by a truth assignment τ ($\tau(c) = 1$) if $\tau(x) = 1$ for a literal x in c, or $\tau(x) = 0$ for a literal $\neg x$ in c; otherwise τ does not satisfy c ($\tau(c) = 0$).

An instance $\varphi = (\varphi_h, \varphi_s)$ of the *partial MaxSAT* problem consists of a set φ_h of *hard* clauses and a set φ_s of *soft* clauses. Any truth assignment τ that satisfies every clause in φ_h is a *solution* to φ. The *cost* of a solution τ to φ is $\textsc{cost}(\varphi, \tau) = \sum_{c \in \varphi_s} (1 - \tau(c))$, i.e., the number of soft clauses not satisfied by τ. A solution τ is *optimal* for φ if $\textsc{cost}(\varphi, \tau) \leq \textsc{cost}(\varphi, \tau')$ holds for any solution τ' to φ. Given φ, the partial MaxSAT problem asks to find an optimal solution to φ. From here on, we refer to partial MaxSAT simply as MaxSAT.

3. Extension Enforcement

We continue by recalling definitions of enforcement operators central to this work (Baumann, 2012b; Coste-Marquis et al., 2015) and by formally defining argument-fixed extension enforcement (Coste-Marquis et al., 2015) as the optimization problem considered in this work.

Formally, the task of extension enforcement is to modify the attack structure R of an AF $F = (A, R)$ in a way that a given set T becomes (a subset of) an extension under a given semantics σ. *Strict* enforcement requires that the given set of arguments has to be a σ-extension, while in *non-strict* enforcement it is required to be a subset of a σ-extension. We denote strict by s and non-strict by ns.

Denote by

$$\mathit{enf}(F, T, s, \sigma) = \{R' \mid F' = (A, R'),\ T \in \sigma(F')\}$$

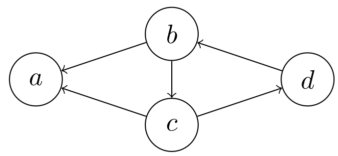
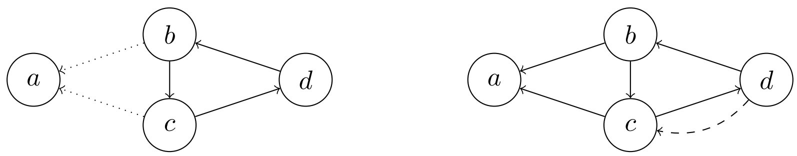


Figure 2: Argumentation framework from Example 3.



(a) Strictly enforcing $\{a\}$. (b) Non-strictly enforcing $\{a\}$.

Figure 3: Enforcement under complete semantics in Example 3.

the set of attack structures that strictly enforce T under σ for an AF F, and analogously

$$enf(F, T, ns, \sigma) = \{R' \mid F' = (A, R'),\ \exists T' \in \sigma(F')\ :\ T' \supseteq T\}$$

for non-strict enforcement.

The Hamming distance between two attack structures R and R' is $|R\Delta R'| = |R \setminus R'| + |R' \setminus R|$, i.e., the cardinality of the symmetric difference, or the number of changes (additions or removals of attacks) of an enforcement. In this work we consider extension enforcement as an optimization problem, where the number of changes is minimized. Formally, the problem can be stated as follows.

Optimal Extension Enforcement ($M \in \{s, ns\}$)
Input: AF $F = (A, R)$, $T \subseteq A$, and semantics σ.
Task: Find an AF $F^* = (A, R^*)$ with

$$R^* \in \underset{R' \in enf(F,T,M,\sigma)}{\arg\min} |R\Delta R'|.$$

Example 3. *Let $F = (A, R)$ be an AF with $A = \{a, b, c, d\}$ and $R = \{(b, a), (b, c), (c, a), (c, d), (d, b)\}$. The corresponding graph representation is shown in Figure 2. Consider enforcing $\{a\}$ under the complete semantics. Note that for F we have $com(F) = \{\emptyset\}$, i.e., the only complete extension is the empty set. A way to strictly enforce $\{a\}$ as a complete extension is to remove the attacks (b, a) and (c, a) (Figure 3a). Adding (d, c) makes $\{a, d\}$ a complete extension, and thereby $\{a\}$ becomes non-strictly enforced (Figure 3b).*

For non-strict extension enforcement, the problems of enforcing an extension under the admissible, complete and preferred semantics coincide.

Theorem 1. *Let $F = (A, R)$ be an AF, $T \subseteq A$. Now*

$$enf(F, T, ns, adm) = enf(F, T, ns, com) = enf(F, T, ns, prf).$$

Table 1: Complexity results for argument-fixed extension enforcement.

σ	strict	non-strict
Conflict-free	in P	in P
Naive	in P	in P
Admissible	in P	NP-complete
Stable	in P	NP-complete
Complete	NP-complete	NP-complete
Grounded	NP-complete	NP-complete
Preferred	Σ_2^P-complete	NP-complete
Semi-stable	Σ_2^P-complete	Σ_2^P-complete
Stage	coNP-hard and in Σ_2^P	Σ_2^P-complete

Proof. First, suppose $R' \in enf(F, T, ns, prf)$, and let $F' = (A, R')$. Since now there exists a superset $T' \supseteq T$ such that $T' \in prf(F')$, and in addition $prf(F') \subseteq com(F')$, it also holds that $T' \in com(F')$. Therefore $R' \in enf(F, T, ns, com)$, which shows that $enf(F, T, ns, prf) \subseteq enf(F, T, ns, com)$ holds. The inclusion $enf(F, T, ns, com) \subseteq enf(F, T, ns, adm)$ is handled similarly by noting that complete extensions of the new AF are a subset of the admissible extensions.

Now, suppose $R' \in enf(F, T, ns, adm)$, and let again $F' = (A, R')$. From the definition we know that there exists $T' \supseteq T$ such that $T' \in adm(F')$. Let now $T'' \in adm(F')$ be a maximal element with respect to set-inclusion such that $T' \subseteq T''$. It follows that $T'' \in prf(F')$, and since $T \subseteq T''$, we know that $R' \in enf(F, T, ns, prf)$. Therefore the final inclusion $enf(F, T, ns, adm) \subseteq enf(F, T, ns, prf)$ also holds, establishing equality between the three sets. $\square$

The fact that the optimal solutions are also the same for non-strict extension enforcement under the admissible, complete, and preferred semantics follows trivially from Theorem 1.

4. Complexity Analysis

This section is dedicated to complexity analysis of non-strict and strict extension enforcement under various central AF semantics. In the decision problem for extension enforcement, we are given an AF $F = (A, R)$, a semantics σ, a set $T \subseteq A$, and an integer $k \geq 0$, and are asked to decide if there is an $F' = (A, R')$ with $|R \Delta R'| \leq k$ that enforces T non-strictly (resp. strictly) for semantics σ. That is, the decision problem for extension enforcement asks whether a given set of arguments can be enforced non-strictly (strictly) with at most k modifications to the attack structure.

An overview of the complexity results of this paper is given in Table 1. Contrasting with the central (non-dynamic) credulous acceptance problem, where we are given an AF $F = (A, R)$, an argument $a \in A$, and a semantics σ, and asked to decide whether $a \in \bigcup \sigma(F)$, we note that under the grounded semantics, both variants of enforcement are NP-complete, while credulous acceptance is decidable in polynomial time. For all other semantics considered here, the complexities of non-strict enforcement and credulous acceptance coincide (Dimopoulos & Torres, 1996; Coste-Marquis, Devred, & Marquis, 2005; Dunne & Caminada, 2008; Dvořák & Woltran, 2010). On the other hand, strict enforcement is computable in polynomial time under the admissible and stable semantics, and credulous acceptance is NP-complete (Dimopoulos & Torres, 1996). Additionally, strict enforcement under the preferred semantics is Σ_2^P-complete, and credulous acceptance is NP-complete, since credulous acceptance under the admissible and preferred semantics, respectively, coincide.

We begin our analysis with non-strict extension enforcement. A basic observation is that, to enforce a set T under semantics σ, all attacks "inside" T need to be removed, since all considered semantics are based on conflict-free sets. For non-strict enforcement under conflict-free and naive semantics, this modification turns out to be optimal.

Theorem 2. *Non-strict extension enforcement for conflict-free and naive semantics is in* P.

Proof. Let $F = (A, R)$ be an AF and $T \subseteq A$ the set to be enforced. Define $F^* = (A, R^*)$ with $R^* = R \setminus (T \times T)$. Now $T \in cf(F^*)$ and thus there is a $T' \in naive(F^*)$ with $T \subseteq T'$. For any $R' \subseteq A \times A$ with $|R \Delta R'| < |R \Delta R^*|$ it holds that T and all supersets of T are not conflict-free in $F' = (A, R')$. Thus F^* is the unique optimal solution. □

For the remaining semantics, non-strict enforcement is presumably harder. This follows from the fact that it is computationally hard to check whether there is a superset of T that is a σ-extension of the input AF F.

Theorem 3. *Non-strict extension enforcement for admissible, complete, preferred, and stable semantics is* NP*-complete.*

Proof. Recall that non-strict extension enforcement under the admissible, complete and preferred semantics coincide (Theorem 1). Therefore it is enough to consider one of these, for instance the admissible semantics. Let $F = (A, R)$ be an AF, $T \subseteq A$ the set to be enforced, $\sigma \in \{adm, stb\}$, and $k \geq 0$ an integer. Let R' be a nondeterministic guess for the attack structure of the proposed solution AF $F' = (A, R')$, and T' the guess for the superset of T that is a σ-extension. Since the necessary checks $|R \Delta R'| \leq k$, $T \subseteq T'$ and $T' \in \sigma(F')$ can be computed in polynomial time, it holds that non-strict extension enforcement under these semantics is in NP.

NP-hardness follows from a reduction from the NP-complete credulous acceptance problem under semantics σ (Dimopoulos & Torres, 1996), where we have to decide whether an argument a is contained in one σ-extension of a given AF F. This problem is reduced to non-strict extension enforcement as follows. Let $F = (A, R)$ be an AF, and $a \in A$ an argument whose credulous acceptance is to be checked. Define a non-strict extension enforcement instance with the same AF F and the singleton set $T = \{a\}$ to be enforced. The reduction can clearly be computed in polynomial time. Suppose T can be enforced with 0 changes to the attack structure. This implies that a is credulously accepted, since there exists $T' \supseteq T$ such that $T' \in \sigma(F)$. On the other hand, if a is credulously accepted, there exists some $T' \in \sigma(F)$ such that $T \subseteq T'$, that is, T is non-strictly enforced with 0 changes. □

The same complexity jump from NP-completeness of credulous acceptance under admissible, complete, preferred, and stable semantics to Σ_2^P-completeness under semi-stable and stage semantics holds with non-strict extension enforcement, as we show next.

Theorem 4. *Non-strict extension enforcement for semi-stable and stage semantics is Σ_2^P-complete.*

Proof. We show Σ_2^P-completeness for these problems similarly as in the NP-completeness proof of Theorem 3. Let $F = (A, R)$ be an AF, $T \subseteq A$ the set to be enforced, $\sigma \in \{sem, stg\}$, and $k \geq 0$ an integer. Let R' be a nondeterministic guess for the attack structure of the proposed solution

AF $F' = (A, R')$, and T' the guess for the superset of T that is a σ-extension. Since the checks $|R \Delta R'| \leq k$, $T \subseteq T'$ can be computed in polynomial time, and $T' \in \sigma(F')$ via a single call to an NP-oracle, it is clear that non-strict extension enforcement under these semantics is in Σ_2^P.

Σ_2^P-hardness follows from a reduction from the Σ_2^P-complete credulous acceptance problem under semantics σ (Dunne & Caminada, 2008; Caminada, Carnielli, & Dunne, 2012; Dvořák & Woltran, 2010). Let $F = (A, R)$ be an AF, and $a \in A$ an argument whose credulous acceptance is to be checked. Define a non-strict extension enforcement instance with the same AF F and the set $T = \{a\}$ to be enforced. The reduction can clearly be computed in polynomial time. Now T can be enforced with 0 changes to the attack structure if and only if a is credulously accepted. □

Coste-Marquis et al. (2015) established that the *union* of non-strict and strict extension enforcement under stable semantics is NP-hard. As a more fine-grained analysis, by Theorem 3 non-strict enforcement is in itself NP-complete; furthermore, in the following we will show that strict enforcement under stable semantics is in fact in P.

From the previous theorems it might appear that the main source of intractability does not originate from the modifications of the attack structure, but from (credulous) acceptance problems associated with the semantics under consideration. However, even for the grounded semantics, non-strict enforcement turns out to be NP-complete. This suggests that the non-determinism introduced by changes in the attack structure is another source for NP-hardness.

Theorem 5. *Non-strict extension enforcement for grounded semantics is* NP*-complete.*

Proof. Membership follows from a non-deterministic guess of a new attack structure and then computing the grounded extension which can be done in polynomial time. Finally, we check whether the desired set is contained in the grounded extension and whether the cardinality of the symmetric difference between the original and new framework does not exceed the given bound.

For hardness, we provide a reduction from the problem of checking whether a given Boolean formula ϕ in CNF is satisfiable. Let $X = \{x_1, \ldots, x_n\}$ be the vocabulary of ϕ and C the set of clauses of ϕ. We view clauses as sets of literals. For a set X, let $\overline{X} = \{\overline{x} \mid x \in X\}$. We construct an AF $F = (A, R)$ as follows. We define the arguments by

$$\begin{aligned} A = &\{t_1, t_2\} \cup X \cup \overline{X} \cup \\ &\{c^j \mid c \in C, 1 \leq j \leq n+1\} \cup \\ &\{y_i^j, \overline{y}_i^j, z_i^j \mid x_i \in X, 1 \leq j \leq n+1\}, \end{aligned}$$

and the attacks by

$$\begin{aligned} R = &\{(x_i, c^j) \mid x_i \in c, x_i \in X, c \in C, 1 \leq j \leq n+1\} \cup \\ &\{(\overline{x}_i, c^j) \mid \neg x_i \in c, \overline{x_i} \in \overline{X}, c \in C, 1 \leq j \leq n+1\} \cup \\ &\{(c^j, t_1) \mid c \in C, 1 \leq j \leq n+1\} \cup \\ &\{(x_i, \overline{x}_i), (\overline{x}_i, x_i) \mid x_i \in X\} \cup \\ &\{(x_i, y_i^j), (\overline{x}_i, \overline{y}_i^j) \mid x_i \in X, 1 \leq j \leq n+1\} \cup \\ &\{(y_i^j, z_i^j), (\overline{y}_i^j, z_i^j) \mid 1 \leq i \leq n, 1 \leq j \leq n+1\} \cup \\ &\{(z_i^j, t_2) \mid 1 \leq i \leq n, 1 \leq j \leq n+1\}. \end{aligned}$$

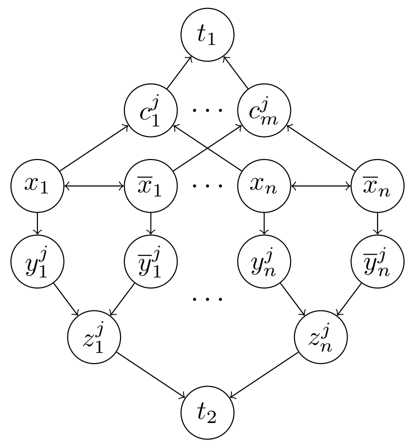


Figure 4: Part of the reduction for Theorem 5 for one value of j (full reduction includes "copies" for $1 \leq j \leq n+1$) with clauses $c_1 = (x_1 \vee x_n)$ and $c_m = (\neg x_1 \vee \neg x_n)$.

The constructed AF, restricted to one value of j, is shown in Figure 4. We claim that ϕ is satisfiable iff there is an AF $F' = (A, R')$ such that $T' \in grd(F')$, $T \subseteq T'$, and $|R \Delta R'| \leq n$, where $n = |X|$ and $T = \{t_1, t_2\}$.

Assume that ϕ is satisfiable. Then there is a truth assignment τ such that $\tau \models \phi$. Define the shorthands $X_1 = \{x_i \in X \mid \tau(x_i) = 1\}$ and $X_0 = \{x_i \in X \mid \tau(x_i) = 0\}$. We define $R' = R \setminus D$ with $D = \{(\overline{x}_i, x_i) \mid x_i \in X_1\} \cup \{(x_i, \overline{x}_i) \mid x_i \in X_0\}$. In words, we remove attacks from arguments in $\overline{X_1}$ onto arguments in X_1 if the corresponding truth value is 1, and from X_0 onto $\overline{X_0}$ if the corresponding truth value is 0. For each pair x_i and $\overline{x}_i$ we remove one direction of the mutual attacks between them. It holds that $|D| = n$. We now show that it holds that $T \subseteq G$ with G the grounded extension of $F' = (A, R')$. Since there is no attack in F' onto arguments in X_1 and $\overline{X_0}$, these arguments are all in the grounded extension of F'. Since these are the only unattacked arguments in F' it holds that $(X_1 \cup \overline{X_0}) = \mathcal{F}_{F'}(\emptyset)$. For each i with $1 \leq i \leq n$ it holds that exactly one of $\{y_i^j, \overline{y}_i^j\}$ for each j with $1 \leq j \leq n+1$ is defended by $\mathcal{F}_{F'}(\emptyset)$. If $x_i \in \mathcal{F}_{F'}(\emptyset)$, then $\overline{y}_i^j$ is defended, otherwise if $\overline{x}_i \in \mathcal{F}_{F'}(\emptyset)$, then y_i^j is defended. This means that $\{y_i^j, \overline{y}_i^j\} \cap \mathcal{F}_{F'}^2(\emptyset) \neq \emptyset$ for each i, j with $1 \leq i \leq n$ and $1 \leq j \leq n+1$. This implies each z_i^j is attacked by the grounded extension and in turn t_2 is defended by the grounded extension, implying that t_2 is part of the grounded extension. Finally, t_1 is defended by $\mathcal{F}_{F'}(\emptyset)$, since each c^j with $c \in C$ is attacked by $\mathcal{F}_{F'}(\emptyset)$ due to the assumption that $\tau \models \phi$. This can be seen due to the fact that for each $c \in C$ there exists a literal $l \in C$ such that $\tau \models l$. If $l = x_i$, then $x_i \in X_1$ and $x_i \in G$. Otherwise, if $l = \neg x_i$, then $\overline{x_i} \in \overline{X_0}$ and $\overline{x_i} \in G$. In both cases the corresponding attacking arguments for c^j are in the grounded extension. Therefore also t_1 is in the grounded extension.

For the other direction assume that T is non-strictly enforced by $F' = (A, R')$ such that $T \subseteq G$ with G the grounded extension of F', and at most n changes are made from the original F, i.e., $|R \Delta R'| \leq n$. We first prove some auxiliary results. We first argue that for each $x_i \in X$ at least one of $\{x_i, \overline{x}_i\}$ is attacked by the grounded extension of F'. For showing that consider $R_j = \{(a, b) \mid a \in A, b \in \{y_i^j, \overline{y}_i^j, z_i^j\}\} \cup \{(z_i^j, t_2)\}$. That is, R_j contains all attacks from all arguments onto $\{y_i^j, \overline{y}_i^j, z_i^j\}$ and the single attack from z_i^j to t_2. For each j, j' with $1 \leq j, j' \leq n+1$ it holds that $R_j \cap R_{j'} = \emptyset$ for $j \neq j'$. Now, we prove that there exists a j such that $R \cap R_j = R' \cap R_j$, that is, there is no change between F and F' on the set R_j. Since $|R \Delta R'| \leq n$, there must exist such a j by a simple pigeon-hole argument (there are at most n changes, but we have $n+1$ such R_j). Now let j be such that $R \cap R_j = R' \cap R_j$. Since t_2 is in the grounded extension of F', one of $\{y_i^j, \overline{y}_i^j\}$ must be too, since otherwise t_2 would not be defended against z_i^j. This in turn means that one of $\{x_i, \overline{x}_i\}$ must be attacked by the grounded extension of F' (these two arguments attack each one of $\{y_i^j, \overline{y}_i^j\}$).

We now show that the arguments in $X \cup \overline{X}$ which are in the grounded extension of F' form a satisfying assignment of ϕ. Let $R'_j = \{(a, b) \mid a \in A, b \in \{c^j \mid c \in C\}\} \cup \{(c^j, t_1) \mid c \in C\}$. By a similar pigeon-hole argument as before, it holds that there exists a j such that $R \cap R'_j = R' \cap R'_j$. Let j now be such that this equation holds. Since t_1 is in the grounded extension of F', each c^j with $c \in C$ is attacked by the grounded extension. Since c^j is only attacked by arguments in $X \cup \overline{X}$, we construct a partial truth assignment for ϕ by

$$\tau(x_i) = \begin{cases} 1 \text{ if } x_i \in G \cap X \\ 0 \text{ if } \overline{x_i} \in G \cap \overline{X}. \end{cases}$$

From the previous observation it holds that these arguments form a well-defined truth assignment, i.e., for no i it holds that both x_i and $\overline{x_i}$ are in the grounded extension or are assigned both 1 and 0 by τ. This truth assignment may be partial. We now show that $\phi[\tau]$ is a tautology. By $\psi[\tau]$ we denote the formula ψ' where every variable that is defined in τ is replaced by the corresponding truth constant. Then, if $\psi[\tau]$ is a tautology (unsatisfiable), this means that every completion of τ to all variables satisfies (does not satisfy) ψ. Suppose $\phi[\tau]$ is not a tautology. Then $\exists \tau'$ defined on all variables of ϕ such that $\tau' \not\models \phi$ with $\tau'(x) = \tau(x)$ if τ is defined on x. It holds that $\exists c \in C$ s.t. $\tau' \not\models c$. Let $l \in c$. We have that

$$\begin{aligned} &\tau'(l) = 0 \\ &\text{iff } l = x_i \text{ and } \tau'(x_i) = 0 \text{ or } l = \neg x_i \text{ and } \tau'(x_i) = 1 \\ &\text{only if } (x_i, c^j) \in R' \text{ and } x_i \notin G \text{ or } (\overline{x_i}, c^j) \in R' \text{ and } \overline{x_i} \notin G. \end{aligned}$$

If $\tau' \not\models l$ for all $l \in c$ then t_1 is not defended by the grounded extension of F'. This is a contradiction. Thus $\phi[\tau]$ is a tautology and ϕ is satisfiable. $\square$

We move on to strict enforcement. Here we establish polynomial-time results for conflict-free and naive semantics, and, in contrast to non-strict extension enforcement, also for stable and admissible semantics.

Theorem 6. *Strict extension enforcement for conflict-free, naive, admissible, and stable semantics is in* P.

Proof. Let $F = (A, R)$ be an AF, $T \subseteq A$ the set to be enforced, and $\sigma \in \{cf, naive, adm, stb\}$. Suppose $T \neq \emptyset$, since otherwise the problem is trivial. Let $t \in T$ be an arbitrary argument in the set.

1. $\sigma = cf$: Let $R^*_{cf} = R \setminus (T \times T)$, that is, R^* is formed by deleting all attacks where both arguments are in T.

2. $\sigma = naive$: Let

$$R^*_{naive} = (R \setminus (T \times T)) \cup \{(a,a) \mid a \in A \setminus T, \nexists (b,a) \in R, \nexists (a,b) \in R \text{ with } b \in T \cup \{a\}\},$$

that is, add a self-attack to arguments $a \in A \setminus T$ where $T \cup \{a\}$ would be conflict-free otherwise.

3. $\sigma = adm$: Let

$$R^*_{adm} = (R \setminus (T \times T)) \cup \{(t,a) \mid a \in A \setminus T, \exists (a,b) \in R \colon b \in T, \nexists (c,a) \in R \colon c \in T\},$$

that is, remove all attacks inside T and add attacks to ensure admissibility by defending from all attacks outside T.

4. $\sigma = stb$: Let $R^*_{stb} = (R \setminus (T \times T)) \cup \{(t,a) \mid a \in A \setminus T, \nexists b \in T \colon (b,a) \in R\}$, removing all attacks inside T and extending the range to arguments not attacked by T.

Now for each of these semantics σ, $F^* = (A, R^*_\sigma)$ is a solution AF where $T \in \sigma(F^*)$. These solutions are also optimal, since in each case no less modifications can be made to the original attack structure. □

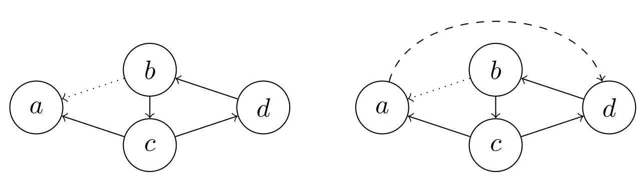
Example 4. *Consider the AF $F = (A, R)$ in Example 3 from page 7. Suppose we want to enforce $T = \{a, b\}$ strictly.*

1. *Under $\sigma = cf$, we need to remove the attack (b, a) in order to make T conflict-free, yielding the AF F^{cf} illustrated in Figure 5a.*

2. *Under $\sigma = naive$, it also suffices to remove the attack (b, a), since in the resulting AF T is already a subset-maximal conflict-free set.*

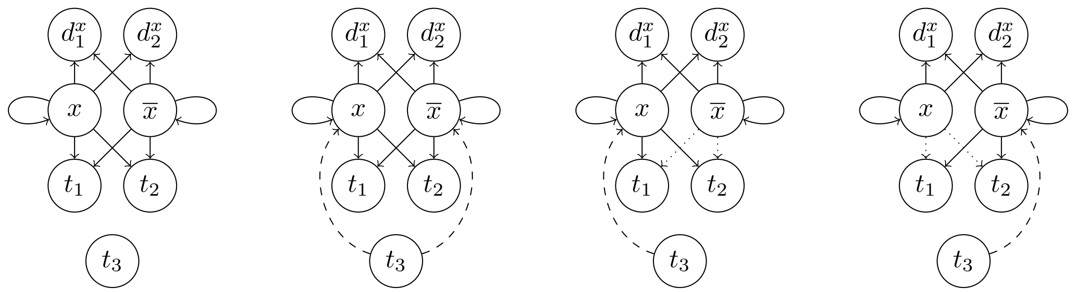
3. *Under $\sigma = adm$, in addition to removing the attack inside T, we need a counterattack on the argument d, since it attacks b and neither a or b attacks it. This is accomplished via adding the attack (a, d), forming the AF F^{adm} illustrated in Figure 5b.*

4. *Under $\sigma = stb$, T must be conflict-free and the range of T must be the whole set of arguments A. Therefore we must delete the attack (b, a), and add an attack from T to argument d, e.g., (a, d). The resulting AF F^{stb} is shown in Figure 5b.*



(a) $F^{cf} = F^{naive}$ (b) $F^{adm} = F^{stb}$

Figure 5: Optimal solutions to strict extension enforcement in Example 4.



(a) Basic structure. (b) Defending d_1^x, d_2^x. (c) Choice x. (d) Choice $\overline{x}$.

Figure 6: Choice gadget for hardness proofs under complete-based semantics.

In contrast to the admissible semantics, strict extension enforcement for complete and grounded semantics is NP-complete. Intuitively, admissibility together with the fact that we must not defend arguments outside any desired set can be used for reducing the NP-complete problem of Boolean satisfiability to strict extension enforcement under complete or grounded semantics.

We present in the following a main part of the reduction that "simulates" truth value assignments in strict extension enforcement for complete-based semantics.

Definition 4. *Let X be a set. We define a function $choice(X) = F$ that returns an AF $F = (A, R)$ as follows with $T = \{t_1, t_2, t_3\}$.*

- $A = X \cup \overline{X} \cup \{d_1^x, d_2^x \mid x \in X\} \cup T$*; and*
- $R = \{(z, t), (z, d_1^x), (z, d_2^x), (z, z) \mid z \in X \cup \overline{X}, t \in \{t_1, t_2\}\}$.

We illustrate this "choice gadget" for a singleton set $X = \{x\}$ in Figure 6a. Enforcing T to be admissible and at the same time not defending one of the $\{d_1^x, d_2^x\}$ arguments, i.e., T to be a complete extension, with at most three changes per element in X can only be achieved by "choosing" either to modify the framework as shown in Figure 6c or in Figure 6d, modulo that the new attack may come from an arbitrary argument in T. This leads to a new attack structure such that T attacks either x or $\overline{x}$, which naturally leads to a truth value assignment over X ("assigning" x to be 1 or 0). Intuitively, with less than two modifications one cannot enforce T to be admissible, since there are four attacks from two arguments (x and $\overline{x}$) onto T. With two modifications, one can enforce T to be admissible (see Figure 6b), but only by making T attacking both x and $\overline{x}$, thereby defending d_1^x and d_2^x, implying that T is not complete ($\mathcal{F}_F(T) = T \cup \{d_1^x, d_2^x\}$). With three modifications one can strictly enforce T to be admissible and not defending either d_1^x or d_2^x, by attacking either x or $\overline{x}$ and removing attacks from the argument not counter-attacked. These properties of $choice(X) = F$ also hold if F is part of a larger AF, if there are no further attacks onto F from the larger embedding AF.

The next lemma formalizes these observations; a full proof is provided in Appendix A.

Lemma 7. *Let X be a set, $choice(X) = F_g = (A_g, R_g)$, and $T = \{t_1, t_2, t_3\}$ with $T \subseteq A_g$. Further, let $F = (A, R)$ be an AF such that $A_g \subseteq A$, $R_g \subseteq R$, and $\nexists (a, a') \in R$ with $a' \in A_g$ and $a \in A \setminus A_g$. We define*

$$\begin{aligned}\hat{F} = \{F' \mid & F' = (A, R'), T \subseteq \mathcal{F}_{F'}(T), T \in cf(F'), \\ & \mathcal{F}_{F'}(T) \cap \{d_1^x, d_2^x \mid x \in X\} = \emptyset, |R \Delta R'| \leq 3 \cdot |X|\}.\end{aligned}$$

For each $F' \in \hat{F}$ with $F' = (A, R')$ and each $x \in X$ exactly one of the following statements holds.

1. *There exists a $(t, x) \in R'$ with $t \in T$, and there does not exist an $(\overline{x}, t') \in R'$ with $t' \in T$.*
2. *There exists a $(t, \overline{x}) \in R'$ with $t \in T$, and there does not exist an $(x, t') \in R'$ with $t' \in T$.*

Further, it holds that $(R \Delta R' \subseteq (T \cup X \cup \overline{X}) \times (T \cup X \cup \overline{X}))$.

Equipped with this intermediate result, we are ready to show NP-completeness of strict extension enforcement for complete and grounded semantics.

Theorem 8. *Strict extension enforcement for complete and grounded semantics is* NP*-complete.*

Proof. Regarding membership, for a given $F = (A, R)$, an integer k, and set T to be enforced, we guess an $F = (A, R')$ and verify in polynomial time whether $T \in com(F')$ or $T \in grd(F')$, and $|R \Delta R'| \leq k$.

For hardness we reduce from the NP-complete problem of checking whether a given conjunctive normal form propositional formula ϕ is satisfiable. Let X be the vocabulary of ϕ, C the set of clauses in ϕ. Let $F_g = (A_g, R_g) = choice(X)$ with $T = \{t_1, t_2, t_3\}$. We construct an instance of the strict enforcement problem as follows.

$$\begin{aligned} A =& A_g \cup C, \\ R =& R_g \cup \{(x, c) \mid x \in X, c \in C, x \in c\} \cup \\ & \{(\overline{x}, c) \mid \overline{x} \in \overline{X}, c \in C, \neg x \in c\}. \end{aligned}$$

Further, we define $k = 3 \cdot |X|$. AF $F = (A, R)$ can be constructed in polynomial time. See Figure 7 for an illustration. Let $\sigma \in \{com, grd\}$. We now claim T can be enforced in F under σ with at most k modifications iff ϕ is satisfiable.

Assume ϕ is satisfiable. Then there is a truth assignment τ such that $\tau \models \phi$. We define the changes in attack relation by D_1 and D_2 as follows.

$$\begin{aligned} D_1 =& \{(x, t) \mid \tau(x) = 1, t \in \{t_1, t_2\}\} \cup \\ & \{(\overline{x}, t) \mid \tau(x) = 0, t \in \{t_1, t_2\}\}, \\ D_2 =& \{(t_3, x) \mid \tau(x) = 0\} \cup \\ & \{(t_3, \overline{x}) \mid \tau(x) = 1\}. \end{aligned}$$

Now let $R' = (R \setminus D_1) \cup D_2$. In words, we add attacks onto $\overline{x}$ if x is assigned true in the satisfying assignment of ϕ (onto x if x is assigned 0 in the satisfying assignment of ϕ). Further, we remove attacks from x to T (from $\overline{x}$ onto T) if x is assigned 1 (0). It holds that $|R \Delta R'| = 3 \cdot |X|$, since for each $x \in X$ we add one attack to, resp. remove two attacks from, F' compared to F.

The set T defends itself in F'. This can be seen by the fact that for each $x \in X$ and $\overline{x} \in \overline{X}$ either there is no attack onto T or T attacks the attacking argument. Further, $D_x = \{d_1^x, d_2^x\}$ is not defended by T in F', since one of $\{x, \overline{x}\}$ is not attacked by T, but attacks both arguments in D_x. Further, $\mathcal{F}_{F'}(\emptyset) = \{t_3\}$, and $\mathcal{F}_{F'}(\{t_3\}) \supseteq \{t_1, t_2\}$, i.e., the grounded extension of F' contains T. What remains to be shown is that $C \cap \mathcal{F}_{F'}(T) = \emptyset$, i.e., T is the grounded extension of F'. Suppose

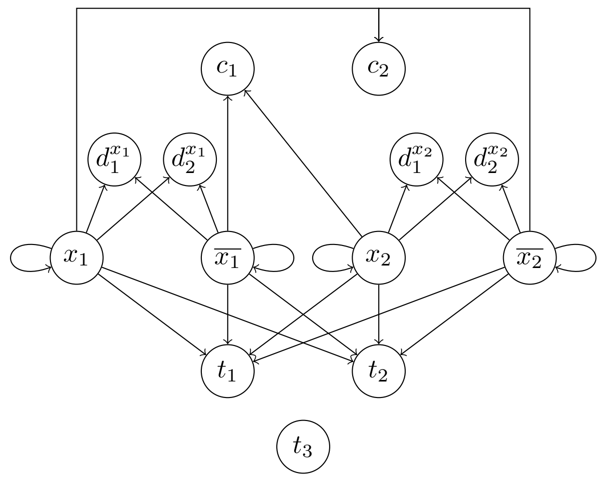


Figure 7: Reduction for Theorem 8 with clauses $c_1 = (\neg x_1 \vee x_2)$ and $c_2 = (x_1 \vee \neg x_2)$.

that there exists a $c \in C$ such that $c \in \mathcal{F}_{F'}(T)$. It holds that

$$\begin{aligned}
&c \in \mathcal{F}_{F'}(T) \\
\text{iff } &\forall (z, c) \in R', \exists t \in T \text{ s.t. } (t, z) \in R' \\
\text{iff } &\forall l \in c, \exists t \in T \text{ s.t.} \\
&\quad \text{if } l = x \text{ then } (x, c) \in R' \text{ and } (t, x) \in R' \\
&\quad \text{and if } l = \neg x \text{ then } (\overline{x}, c) \in R' \text{ and } (t, \overline{x}) \in R' \\
\text{iff } &\forall l \in c \\
&\quad \text{if } l = x \text{ then } \tau(x) = 0 \\
&\quad \text{and if } l = \neg x \text{ then } \tau(x) = 1 \\
\text{iff } &\tau \not\models c.
\end{aligned}$$

Therefore, T defends an argument $A \setminus T$ iff τ is not a satisfying assignment of ϕ. By assumption that $\tau \models \phi$, it holds that T is the grounded extension of F' (and a complete extension of F').

Now we show the other direction. It suffices to consider the case with $\sigma = com$ (the grounded extension is also a complete extension). Assume that there exists an $F' = (A, F')$ such that $T \in com(T)$ and $|R \Delta R'| \leq 3$. Due to Lemma 7 we can immediately infer that exactly one of the two statements in that lemma hold for each $x \in X$ regarding F'. Further, due to the same lemma, the modifications specified in these two statements are the only changes in the new AF. Define a truth assignment τ as follows.

$$\tau(x) = \begin{cases} 1 \text{ if } \nexists t \in T \text{ s.t. } (t, x) \in R' \\ 0 \text{ if } \nexists t \in T \text{ s.t. } (t, \overline{x}) \in R' \end{cases}$$

We now claim that $\tau \models \phi$. Suppose the contrary, that is, there exists a clause c in ϕ such that $\tau \not\models c$. This means that if $x \in c$, then $\tau(x) = 0$ and if $\neg x \in c$ then $\tau(x) = 1$. This in turn implies that all attackers of c in F' are attacked by T and T defends c, implying that T is not complete in F', which is a contradiction. $\square$

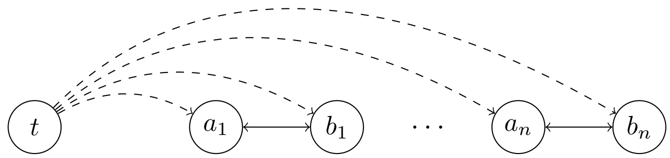


Figure 8: Strict enforcement under preferred semantics.

For preferred and semi-stable semantics, we see a jump in complexity: the corresponding problems are in fact Σ_2^P-complete. Intuitively, in addition to the source of intractability that strict enforcement under complete semantics brings, one has to take into account that modifications to the attack structure might give rise to supersets of T that are admissible. We show Σ_2^P-hardness by a reduction from satisfiability of quantified Boolean formulas. The reduction utilizes the same choice gadget (Definition 4) as the hardness proof for strict extension enforcement under complete and grounded semantics, but, additionally, incorporates argument and attack constructions to represent universally quantified variables. The full proof is provided in Appendix A.

Theorem 9. *Strict extension enforcement for preferred and semi-stable semantics is Σ_2^P-complete.*

As an example of when complete and preferred semantics differ w.r.t. strict enforcement, see Figure 8. Here strictly enforcing $\{t\}$ under complete semantics requires no modifications. In contrast, under preferred semantics each other argument requires one distinct modification.

Finally, for stage semantics we give straightforward bounds. Hardness follows from coNP-hardness of verifying if a set is a stage extension (Dvořák & Woltran, 2010).

Corollary 10. *Strict extension enforcement for stage semantics is in Σ_2^P and* coNP-*hard.*

Proof. Membership follows by guessing a new attack structure and verifying if the given set is a stage extension in the new framework (verification is in coNP) and checking whether the number of changes is within the bound.

For hardness, we reduce from the verification problem of stage semantics to strict enforcement. Given an AF $F = (A, R)$ and $S \subseteq A$, we construct an instance of strict enforcement by defining the input AF to be F, the enforced set to be S, and the allowed changes to be 0. Then S is strictly enforceable in F iff $S \in stg(F)$. □

Based on our attempts to pinpoint its complexity, we conjecture that strict enforcement for stage semantics is Σ_2^P-complete; this is the only missing piece in the complexity map (recall Table 1) established in this paper. The same technique as for the Σ_2^P-hardness proof of preferred and semi-stable semantics cannot be applied to stage semantics. For instance, for the AF in Figure 6a (the choice gadget used to prove hardness for complete, grounded, preferred, and semi-stable semantics), the set $S = \{t_1, t_2, t_3, d_1^x, d_2^x\}$ is a stage extension. Enforcing $T = \{t_1, t_2, t_3\}$ strictly to be a stage extension can be achieved by addition of two attacks to make T conflicting with $\{d_1^x, d_2^x\}$. This is in contrast to complete-based semantics, which, if restricted to three modifications, requires changes to attacks between T and $\{x, \overline{x}\}$ to make T admissible and not defend $\{d_1^x, d_2^x\}$.

Each stage extension is a naive extension, yet strict extension enforcement under naive semantics (a problem that can be decided in polynomial time, recall Theorem 6) differs to strict extension

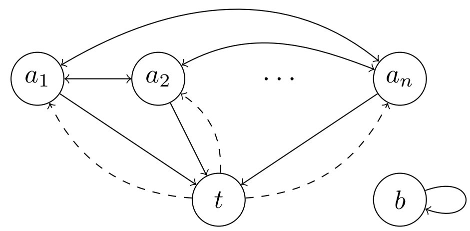


Figure 9: Strict enforcement under stage semantics.

enforcement under stage semantics. This is witnessed by the example in Figure 9, with an AF that has an n-sized clique subgraph (a_i attacks a_j for $1 \leq i, j \leq n$). In this AF, there is no stable extension and each singleton set $\{a_i\}$ is a stage extension. An optimal solution to strictly enforcing $T = \{t\}$ as a stage extension is to have t attack each a_i. In contrast, T is already a naive extension of the original AF. Further, solving strict extension enforcement for stage semantics is different in this example to strict extension enforcement under stable semantics (which can be solved in polynomial time, recall Theorem 6), since b is not attacked by T in the modified AF.

5. Extension Enforcement via MaxSAT

In this section we present declarative encodings that can be used for solving extension enforcement optimally under various semantics. We employ MaxSAT as a well-suited declarative language. The NP-complete problems of strict extension enforcement under the complete semantics, and non-strict extension enforcement under the admissible, complete, preferred and stable semantics are solved via a direct MaxSAT encoding of the problem. For non-strict enforcement under stable semantics, our encoding is essentially the same as the integer programming formulation presented by Coste-Marquis et al. (2015). Note that by Theorem 1, non-strict enforcement under the admissible, complete, and preferred semantics, respectively, coincide. Hence these problems may all be solved using the encoding provided in this work for the admissible semantics. The encoding used to solve each problem is summarized in Table 2. We return to the computationally harder problem variants in the next section.

5.1 Soft Clauses for Optimization

In MaxSAT, soft clauses are used to encode the objective function, i.e., the function over which we are optimizing. Let $F = (A, R)$ be the AF of an enforcement instance. For each $a, b \in A$, define Boolean variables $r_{a,b}$ with the interpretation $\tau(r_{a,b}) = 1$ iff the attack (a, b) is included in the attack structure R' of the solution AF $F' = (A, R')$. The objective function to be minimized is the number of changes to the original attack structure. This can be expressed by the soft clauses

$$\varphi_s(F) = \bigwedge_{a,b \in A} r'_{a,b},$$

where

$$r'_{a,b} = \begin{cases} r_{a,b} & \text{if } (a, b) \in R, \\ \neg r_{a,b} & \text{if } (a, b) \notin R. \end{cases}$$

The above soft clauses are satisfiable if and only if there are no changes to the attack structure. By maximizing the number of clauses satisfied, we are minimizing the Hamming distance between the original and solution AFs.

5.2 Hard Clauses for Extension Enforcement

In addition to the objective function, we need to encode the underlying properties of the problem into the MaxSAT instance. This is accomplished via hard clauses. Let $F = (A, R)$ be an AF and $T \subseteq A$ the set to be enforced under semantics σ. In addition to the attack variables, for each $a \in A$, define Boolean variables x_a with the intended meaning of $\tau(x_a) = 1$ iff a is included in the extension of the new AF F'.

For non-strict extension enforcement, define

$$\textsc{ext}(ns, F, T) = \bigwedge_{a \in T} x_a,$$

which states that T is a subset of a σ-extension.

For conflict-free sets, if an attack is present between two arguments, they cannot occur in the same extension. This is stated as

$$\textsc{ext}(ns, F, T, cf) = \textsc{ext}(ns, F, T) \wedge \bigwedge_{a,b \in A} (r_{a,b} \to (\neg x_a \vee \neg x_b)).$$

For admissible sets, if there is an attack on some argument in an extension, there must be some argument in the same extension that attacks the attacker, i.e., the attacked argument is defended by the extension, encoded as

$$\textsc{ext}(ns, F, T, adm) = \textsc{ext}(ns, F, T, cf) \wedge \bigwedge_{a,b \in A} \left((x_a \wedge r_{b,a}) \to \bigvee_{c \in A} (x_c \wedge r_{c,b}) \right).$$

For the stable semantics, the range of the extension is the whole set of arguments. Therefore, if an argument is not included in the extension, there must be an attack on it from some argument in the extension. This fact is expressed by

$$\textsc{ext}(ns, F, T, stb) = \textsc{ext}(ns, F, T, cf) \wedge \bigwedge_{a \in A} \left(\neg x_a \to \bigvee_{b \in A} (x_b \wedge r_{b,a}) \right).$$

We move on to strict enforcement. Since here the values of the x_a variables are fixed, i.e., $\tau(x_a) = 1$ if $a \in T$, and $\tau(x_a) = 0$ otherwise, there is no need for them in the encoding. Now conflict-free sets can be encoded as

$$\textsc{ext}(s, F, T, cf) = \bigwedge_{a,b \in T} \neg r_{a,b},$$

i.e., all attacks inside the set T are removed.

Admissible sets are encoded by

$$\textsc{ext}(s, F, T, adm) = \textsc{ext}(s, F, T, cf) \wedge \bigwedge_{a \in T} \bigwedge_{b \in A \setminus T} \left(r_{b,a} \to \bigvee_{c \in T} r_{c,b} \right).$$

Table 2: Encodings and algorithms for the extension enforcement problem.

Type	Semantics	Encoding/Algorithm
ns	adm, com, prf	$\text{EXT}(ns, F, T, adm)$
ns	stb	$\text{EXT}(ns, F, T, stb)$
ns	sem, stg	Algorithm 1
s	adm	$\text{EXT}(s, F, T, adm)$
s	com	$\text{EXT}(s, F, T, com)$
s	stb	$\text{EXT}(s, F, T, stb)$
s	prf, sem, stg	Algorithm 1

For the complete semantics, in addition all defended arguments must be included in the extension. Therefore, there must be an attack on all arguments outside T that is not defended against by T. This is expressed by the formula

$$\text{EXT}(s, F, T, com) = \text{EXT}(s, F, T, adm) \wedge \bigwedge_{a \in A \setminus T} \bigvee_{b \in A} \left(r_{b,a} \wedge \bigwedge_{c \in T} \neg r_{c,b} \right).$$

6. Counterexample-Guided Abstraction Refinement

In this section, we present algorithms for solving enforcement problems beyond NP, i.e., the Σ_2^P-complete problems of strict extension enforcement under the preferred, semi-stable, and stage semantics, and non-strict extension enforcement under the semi-stable and (the possibly Σ_2^P-complete) stage semantics. Motivated by completeness of these problem variants for the second level of the polynomial hierarchy, we employ a general approach called counterexample-guided abstraction refinement (CEGAR) (Clarke et al., 2004, 2003). The basic idea is to start with an NP-abstraction, which is a problem in NP and an overapproximation of the original problem, and to solve it, leading to a candidate solution. Then we use an NP-oracle (such as a SAT solver) iteratively to check whether the candidate is indeed a solution to the original, harder problem by asking the solver for a counterexample. If there are none, we have found an optimal solution and exit the loop, and if there is a counterexample, we refine the abstraction by adding constraints which rule out the counterexample given by the oracle, and continue by solving the abstraction. While there are no guarantees on finding the actual solution before an exponential blow-up of the encoding of the refinement in general, this approach has been shown to be effective in practice in different domains (Janota, Grigore, & Marques-Silva, 2010; Donaldson, Kaiser, Kroening, Tautschnig, & Wahl, 2012).

6.1 CEGAR Algorithm for Extension Enforcement

We provide CEGAR algorithms for solving the strict extension enforcement problem under the preferred, semi-stable, and stage semantics, and for the non-strict extension enforcement problem under the semi-stable and stage semantics. The NP-abstractions used are the admissible or complete semantics for the preferred and semi-stable semantics, and conflict-free sets for stage, since these overapproximate the problem at hand (recall Definition 3). The abstraction is solved using a MaxSAT solver, leading to a candidate solution, i.e., an AF that is an optimal solution to the extension enforcement under the chosen NP semantics. The validity of the candidate AF is checked using a SAT solver by constructing a formula which is unsatisfiable if and only if T has been enforced in the AF, resulting in a refinement of the NP-abstraction in case a counterexample is found.

Let $F = (A, R)$ be an AF, $T \subseteq A$ the set to be enforced under semantics $\sigma \in \{prf, sem, stg\}$, and let $M \in \{ns, s\}$ be the type of enforcement (non-strict or strict). The CEGAR algorithm for solving these problems is presented as Algorithm 1. The first step is to choose the base semantics, denoted by χ, that acts as an overapproximation of the original problem. We choose conflict-free sets for stage, and the admissible or complete semantics for semi-stable and preferred, since these are solvable via a direct MaxSAT call (recall Section 5.2). Next, we construct the MaxSAT clauses for solving the corresponding extension enforcement problem under the semantics χ, and enter the main loop of the algorithm. In the loop, we solve the problem via a MaxSAT solver call, extracting a truth assignment τ. Then we use it to construct the candidate solution AF $F' = (A, R')$, where

$$R' = \{(a, b) \mid a, b \in A, \tau(r_{a,b}) = 1\}.$$

Now we can use a SAT solver to check whether F' is also a solution to the enforcement problem under the semantics σ. Recall that for strict enforcement, we need to check if $T \in \sigma(F')$, and for non-strict enforcement, we check if $T' \in \sigma(F')$, where

$$T' = \{a \in A \mid \tau(x_a) = 1\}.$$

For this SAT solver call, we encode the base semantics following Besnard and Doutre (2004), by defining for each $a \in A$ a Boolean variable y_a (to distinguish from the x_a variables used in the MaxSAT solver call) with the interpretation $\tau(y_a) = 1$ iff a is included in a χ-extension of the AF F'. Now, conflict-free sets are encoded as

$$\textsc{extension}(F', cf) = \bigwedge_{(a,b) \in R'} (\neg y_a \vee \neg y_b),$$

stating that for all attacks in R', either the attacker or the attacked argument is not contained in the conflict-free set. Admissible semantics can be expressed by the clauses

$$\textsc{extension}(F', adm) = \textsc{extension}(F', cf) \wedge \bigwedge_{(b,a) \in R'} \left(y_a \rightarrow \left(\bigvee_{(c,b) \in R} y_c \right) \right),$$

Algorithm 1 CEGAR-based extension enforcement for $\sigma \in \{prf, sem, stg\}$ with $M = s$, and for $\sigma \in \{sem, stg\}$ with $M = ns$.

```
if σ ∈ {prf, sem} then χ ← com else χ ← cf
φ_h ← EXT(M, F, T, σ)
if M = ns then φ_h ← φ_h ∧ RANGE(A)
while true do
    (c, τ) ← MAXSAT(φ_h, φ_s(F))
    result ← SAT(CHECK(M, A, τ, S, σ))
    if result = unsatisfiable then return (c, τ)
    else φ_h ← φ_h ∧ REFINE(τ, M)
```

that is, an admissible extension is conflict-free, and if there is an attack on an argument in the extension, there must be a defending argument in the extension. Finally, we encode complete semantics by

$$\textsc{extension}(F', com) = \textsc{extension}(F', adm) \wedge \bigwedge_{a \in A} \left(\left(\bigwedge_{(b,a) \in R'} \left(\bigvee_{(c,b) \in R'} y_c \right) \right) \rightarrow y_a \right),$$

i.e., complete extensions are admissible, and each argument that is defended is also included in the extension. For preferred semantics, we check whether there is a strict superset of T, which we express via

$$\textsc{superset}(F', T, prf) = \bigwedge_{a \in T} y_a \wedge \bigvee_{a \in A \setminus T} y_a.$$

In the case of semi-stable and stage semantics, we similarly need to search for a superset with respect to the range of T (for strict enforcement) or T' (for non-strict enforcement), which we denote by S. To accomplish this, we define for each $a \in A$ a new Boolean variable y_a^+ which encode the range of each argument as a conjunction of equivalences

$$\textsc{range}(F') = \bigwedge_{a \in A} \left(y_a^+ \leftrightarrow \left(y_a \vee \bigvee_{(b,a) \in R'} y_b \right) \right).$$

Now, under $\sigma \in \{sem, stg\}$, checking whether there is a superset of S with respect to the range is encoded as

$$\textsc{superset}(F', S, \sigma) = \bigwedge_{a \in S^+_{R'}} y_a^+ \wedge \bigvee_{a \in A \setminus S^+_{R'}} y_a^+ \wedge \textsc{range}(F').$$

If such a superset that is a χ-extension exists, the formula

$$\textsc{check}(M, A, \tau, S, \sigma) = \textsc{extension}(F', \sigma) \wedge \textsc{superset}(F', S, \sigma)$$

is satisfiable, providing a counterexample that F' is not an actual solution, that is, T or T' is not a σ-extension in F'.

For the refinement step, we define the shorthand that encodes the attack structure R' of F' via

$$\textsc{attack}(\tau) = \bigwedge_{(a,b) \in R'} r_{a,b} \wedge \bigwedge_{(a,b) \in (A \times A) \setminus R'} \neg r_{a,b}.$$

The current abstraction ψ is in the strict case refined by

$$\textsc{refine}(\tau, s) = \neg\textsc{attack}(\tau),$$

which rules out the attack structure of the candidate solution. In the non-strict case, however, it may be that the attack structure of our candidate solution is indeed correct, but the superset T' of T is not. In other words, there might be a χ-extension T'' that is also a σ-extension in F'. If this is the case, then the range of the set T'' is not a subset of the range of T'. This is ruled out by

$$\textsc{refine}(\tau, ns) = \neg\textsc{attack}(\tau) \vee \bigvee_{a \in A \setminus T'^+_{R'}} x_a^+,$$

where we have defined range variables (Line 3) via

$$\text{RANGE}(A) = \bigwedge_{a \in A} \left(x_a^+ \leftrightarrow \left(x_a \vee \bigvee_{b \in A} (r_{b,a} \wedge x_b) \right) \right).$$

Algorithm 1 will produce an optimal solution to strict enforcement for $\{prf, sem, stg\}$ and non-strict enforcement for $\{sem, stg\}$, respectively, as at each iteration the current abstraction is solved to optimum.

Example 5. *Consider the AF F in Example 3 on page 7, and enforcing $T = \{d\}$ strictly under the preferred semantics. We go through a hypothetical run of the CEGAR algorithm. Start by defining Boolean variables*

$$r_{a,a}, r_{a,b}, r_{a,c}, r_{a,d}, r_{b,a}, r_{b,b}, r_{b,c}, r_{b,d}, r_{c,a}, r_{c,b}, r_{c,c}, r_{c,d}, r_{d,a}, r_{d,b}, r_{d,c}, r_{d,d}.$$

Using complete as the base semantics for the abstraction, we form the hard clauses $\varphi_h =$ $\text{EXT}(s, F, T, com)$ along with the soft clauses $\varphi_s(F)$, and enter the CEGAR loop of Algorithm 1. Suppose the MaxSAT solver returns a truth assignment corresponding to AF $F' = (A, R')$, where

$$R' = \{(a,c), (b,a), (b,c), (c,a), (d,b)\},$$

i.e., the attack (a, c) has been added and the attack (c, d) removed, illustrated in Figure 10.

Now we form a SAT instance with the variables y_a, y_b, y_c, and y_d, and the clauses

$$\begin{aligned} \text{CHECK}(s, A, \tau, T, prf) &= \text{EXTENSION}(F', adm) \wedge \text{SUPERSET}(F', T, prf) \\ &= (\neg y_a \vee \neg y_b) \wedge (\neg y_a \vee \neg y_c) \\ &\quad \wedge (\neg y_b \vee \neg y_c) \wedge (\neg y_b \vee \neg y_d) \\ &\quad \wedge (y_c \to (y_b \vee y_c)) \wedge (y_a \to y_d) \\ &\quad \wedge (y_c \to y_d) \wedge (y_a \to (y_a \vee y_b)) \wedge (y_b \to \bot) \\ &\quad \wedge y_d \wedge (y_a \vee y_b \vee y_c). \end{aligned}$$

This instance is satisfiable via the truth assignment $\tau(y_a) = 1$, $\tau(y_b) = 0$, $\tau(y_c) = 0$, $\tau(y_d) = 1$, corresponding to the admissible extension $\{a, d\}$. Therefore we get a counterexample, and we refine the solution via adding the hard clause

$$\text{REFINE}(\tau, s) = \neg\text{ATTACK}(\tau)$$

and continue in the CEGAR loop. Now suppose the MaxSAT solver suggests the attack structure

$$R'' = \{(b,a), (b,c), (c,a), (c,d), (d,a), (d,b), (d,c)\},$$

corresponding to an AF $F'' = (A, R'')$, where the attacks (d, a) and (d, c) have been added into the framework, illustrated in Figure 10. Again, we form a SAT instance with the variables y_a, y_b,

y_c, and y_d, and the clauses

$$
\begin{aligned}
\text{CHECK}(s, A, \tau, T, prf) &= \text{EXTENSION}(F'', adm) \wedge \text{SUPERSET}(F'', T, prf) \\
&= (\neg y_a \vee \neg y_b) \wedge (\neg y_a \vee \neg y_c) \\
&\quad \wedge (\neg y_a \vee \neg y_d) \wedge (\neg y_b \vee \neg y_c) \\
&\quad \wedge (\neg y_b \vee \neg y_d) \wedge (\neg y_c \vee \neg y_d) \\
&\quad \wedge (y_a \rightarrow y_d) \wedge (y_c \rightarrow y_d) \\
&\quad \wedge (y_a \rightarrow (y_b \vee y_d)) \wedge (y_d \rightarrow (y_b \vee y_d)) \\
&\quad \wedge (y_a \rightarrow y_c) \wedge (y_b \rightarrow y_c) \wedge (y_c \rightarrow y_c) \\
&\quad \wedge y_d \wedge (y_a \vee y_b \vee y_c)
\end{aligned}
$$

Since we must set $\tau(y_d) = 1$ due to the unit clause y_d, we know that we must set $\tau(y_a) = 0$, $\tau(y_c) = 0$ and $\tau(y_d) = 0$ due to the clauses encoding conflict-freeness. Therefore the instance is unsatisfiable, since the clause $y_a \vee y_c \vee y_d$ is not satisfied. Now we know that the AF $F'' = (A, R'')$ is an optimal solution to the enforcement problem.

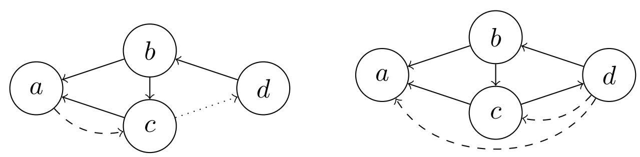


Figure 10: The argumentation frameworks F' (left) and F'' (right) in Example 5.

7. Experiments

We implemented the MaxSAT encodings and CEGAR algorithms described in the previous sections in the software system Pakota (Niskanen, Wallner, & Järvisalo, 2016) available online in open source at `http://www.cs.helsinki.fi/group/coreo/pakota/`. In addition to solving instances of the NP-hard variants of the extension enforcement problem optimally, for the NP-complete problem variants Pakota offers an option for direct output of the produced encoding in standard WCNF and LP file formats for MaxSAT solvers and IP solvers, respectively. This allows for calling any MaxSAT or IP solver on the encoding. Pakota also offers a general interface for plugging in essentially any MaxSAT solver of choice.

Here we present results of an empirical evaluation of the scalability of the Pakota system under different problem variants for extension enforcement, both the NP- and the Σ_2^P-complete ones. In addition we evaluate the impact of the choice of the underlying MaxSAT solver. For the NP-complete problem variants, in the experiments we used the state-of-the-art MaxSAT solvers MaxHS (Davies & Bacchus, 2013), Maxino (Alviano, Dodaro, & Ricca, 2015), MSCG (Morgado, Ignatiev, & Marques-Silva, 2015), Open-WBO (Martins, Manquinho, & Lynce, 2014), and WPM (Ansótegui, Didier, & Gabàs, 2015), using the MaxSAT Evaluation 2015 versions, as well as the commercial IBM CPLEX integer programming solver (version 12.6), by directly solving the CNF encoding as produced by Pakota. For CEGAR, we compare the performance of Open-WBO and LMHS (Saikko, Berg, & Järvisalo, 2016) as the underlying MaxSAT solvers, as supported by

Pakota, and the complete semantics as the base abstraction[2]. Out of the solvers considered, Maxino, MSCG, Open-WBO, and WPM are core-guided MaxSAT solvers, and MaxHS and LMHS are SAT-IP hybrids.

We also compare the performance of Pakota and the only other solver for extension enforcement we are aware of, introduced by Coste-Marquis et al. (2015), which we obtained from the author and developer of the solver, Jean-Guy Mailly. We refer to it as the IJCAI'15 solver. The obtained version of the IJCAI'15 solver does not support any other AF semantics than stable, and therefore no results for other semantics using this solver are presented here.

We generated benchmark instances for the experiments as follows. Given a number of arguments and an edge probability p, we formed an AF based on the Erdős-Rényi random digraph model, where each attack is included independently with probability p. Given an AF and a number of enforced arguments, we constructed a corresponding enforcement instance by sampling the enforced arguments uniformly at random from the set of arguments, without replacement. For each number of arguments $|A| \in \{25, 50, \dots\}$ and each edge probability $p \in \{0.05, 0.1, 0.2, 0.3\}$, we generated five AFs. For each AF, we generated five enforcement instances with $|T|$ enforced arguments, for each $|T|/|A| \in \{0.05, 0.1, 0.2, 0.3\}$. We thus obtained 400 instances for each $|A|$. The generated instances and the benchmark generator are available through the Pakota webpage.

The experiments were run on 2.83-GHz Intel Xeon E5440 quad-core machines with 32-GB RAM and Debian GNU/Linux 8 using a timeout of 900 seconds per instance.

7.1 Results for First-Level Problems

We start with the NP-complete extension enforcement problems of enforcing an extension non-strictly under the admissible and stable semantics, and strictly under the complete semantics. Figures 11 (right), 12 (right) and 13 (right) provide overviews of the median runtimes for Open-WBO, WPM, Maxino, MSCG, MaxHS, and CPLEX with respect to the number of arguments. Open-WBO exhibits better performance than the rest of the solvers in terms of median runtimes. All in all, the core-guided MaxSAT solvers Open-WBO, WPM, Maxino and MSCG perform the best. The plots in Figures 11 (left), 12 (left) and 13 (left) give the number of instances solved (x-axis) as a function of a per-instance time limit (y-axis). CPLEX performs considerably better in terms of solving more instances than the rest of the solvers. However, on non-strict admissible (Figure 11 (left)) and strict complete (Figure 12 (left)) CPLEX is slower than most of the other solvers. As seen from Figure 12 (left), on strict complete CPLEX and Maxino are the only solvers which solve all of the instances generated within the time limit. Furthermore, for strict enforcement under complete semantics, all of the MaxSAT solvers solve more instances compared to non-strict admissible and non-strict stable, indicating that strict complete is empirically easier on these instances.

In Figure 14 (left) we show the median runtimes for Open-WBO and CPLEX on non-strict admissible, strict complete, and non-strict stable extension enforcement, along with the IJCAI'15 solver on non-strict stable. The median runtimes for Open-WBO are clearly lower than for the rest of the solvers, which indicates that using core-guided MaxSAT solvers is a promising approach for solving the extension enforcement problem, with most of the instances even with $|A| = 350$ solved under one second. The performance of the IJCAI'15 solver is essentially the same as that

2. Based on preliminary experimentation, in terms of runtime performance the complete semantics serves as at least as good a base semantics as admissible sets. This follows the natural intuition based on the subset relation between complete and admissible.

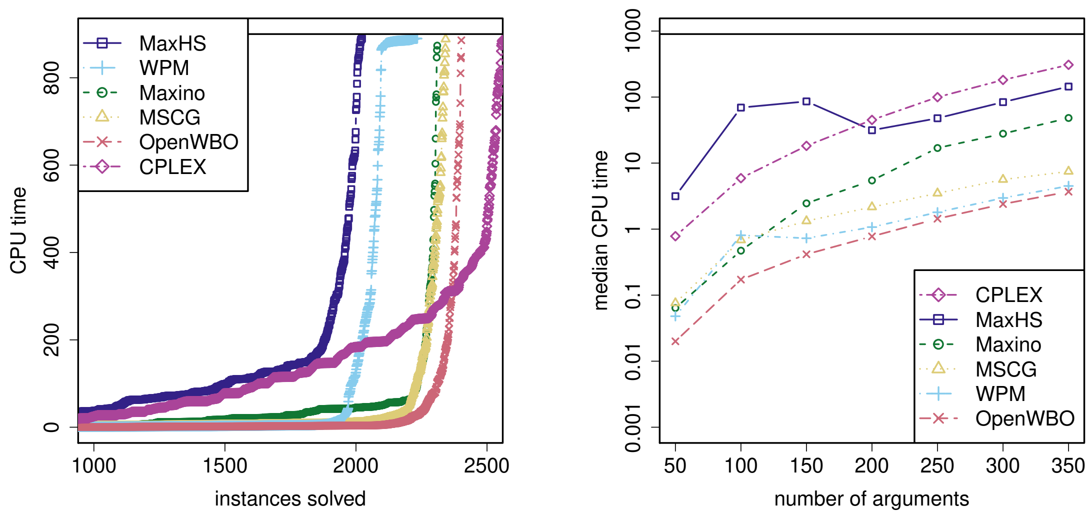


Figure 11: MaxSAT solver comparison for NP-complete non-strict extension enforcement under the admissible semantics.

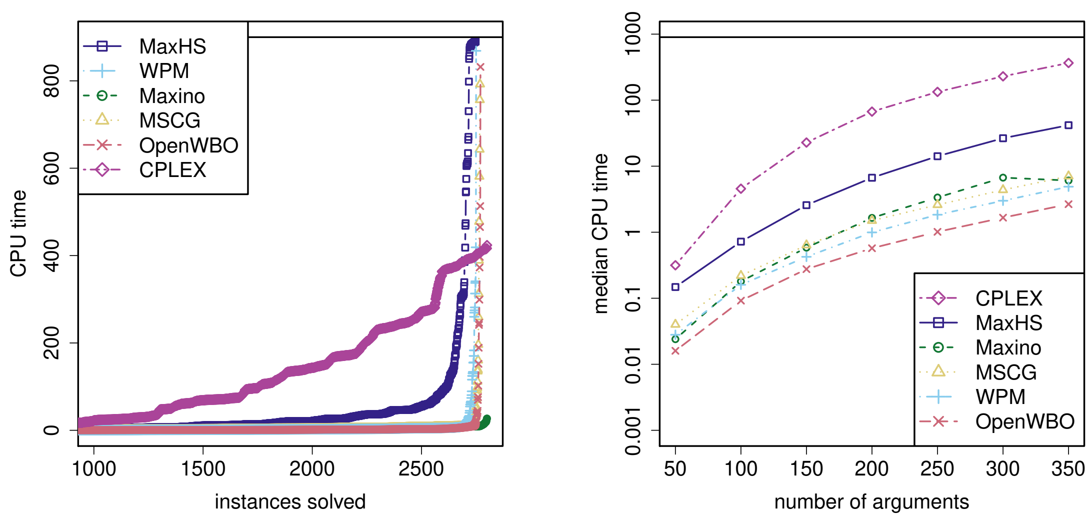


Figure 12: MaxSAT solver comparison for NP-complete strict extension enforcement under the complete semantics.

of CPLEX, which comes as no surprise as it also encodes the problem instance as an IP instance, using CPLEX as the underlying solver.

The relative performance of Open-WBO and CPLEX on non-strict admissible, strict complete, and non-strict stable is illustrated in Figure 14 (right) and Figure 15. For strict complete, Open-WBO solves most instances considerably faster than CPLEX. However, some instances are not solved within the time limit by Open-WBO, as opposed to CPLEX which solves all of the instances within the time limit. For non-strict admissible and non-strict stable, on the other hand, the performance of Open-WBO is not significantly better than that of CPLEX. A large number of instances is solved faster by CPLEX, and CPLEX has only a few timeouts. Non-strict extension enforcement under the admissible semantics seems to be empirically harder for CPLEX than under stable; the distribution of runtimes in Figure 14 (right) is already approaching the time limit at $|A| = 350$.

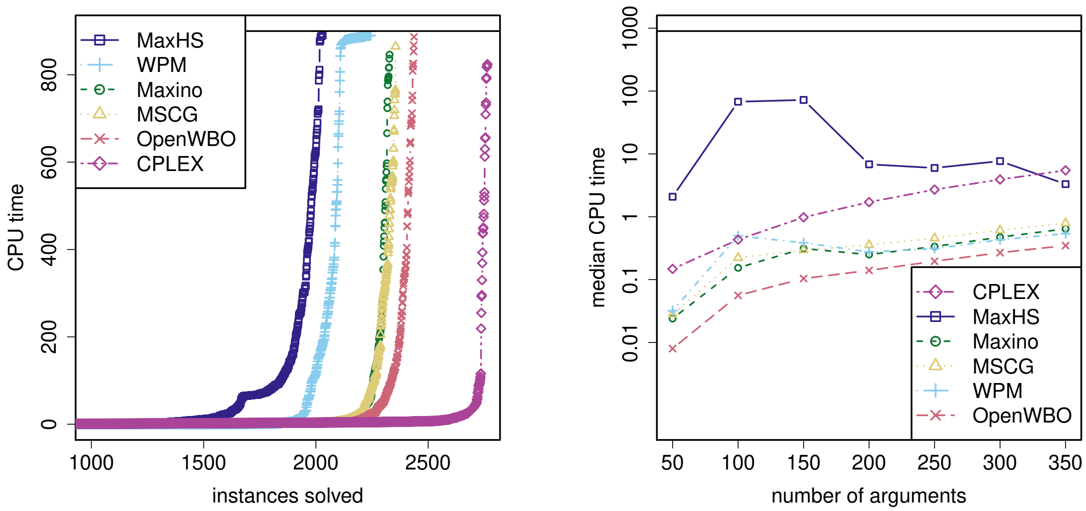


Figure 13: MaxSAT solver comparison for NP-complete non-strict extension enforcement under the stable semantics.

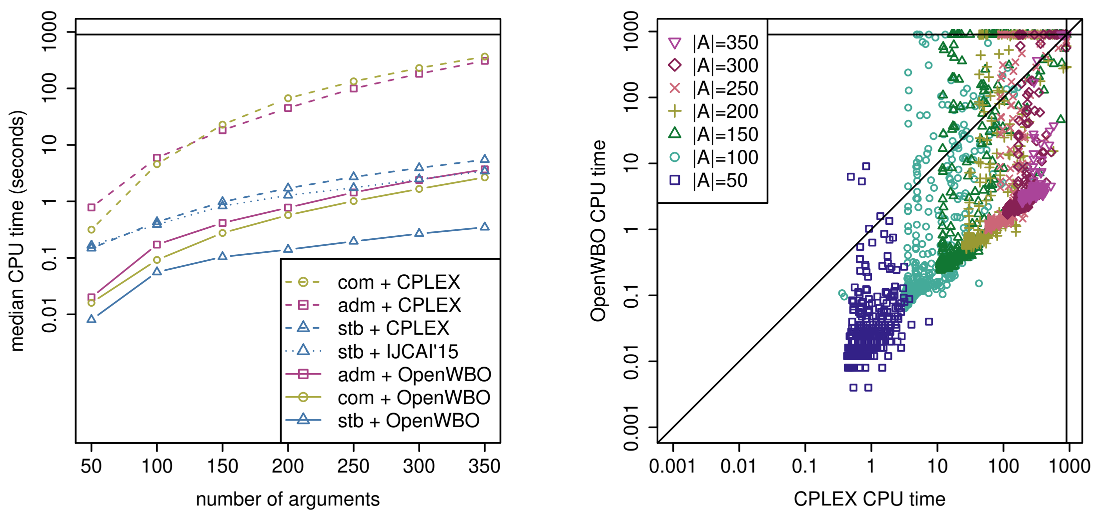


Figure 14: Left: NP-complete extension enforcement (non-strict under admissible and stable, strict under complete); right: OpenWBO versus CPLEX on non-strict admissible.

7.2 Results for Second-Level Problems

Turning to the Σ_2^P-complete problem of enforcing an extension strictly under the preferred semantics, the median runtimes for the CEGAR algorithm are shown in Figure 16 (left) together with the distribution of the runtimes. Even with 175 arguments, most of the instances are solved within one second. Similarly, for the Σ_2^P-complete problem of extension enforcement under the semi-stable semantics, Figure 16 (right) and Figure 17 (left) include plots for the strict and non-strict variants of the problem, on which we observe similar scalability as in the case of preferred semantics.

Figure 17 (right) gives an overview of the relative performance of the MaxSAT solvers Open-WBO and LMHS on the CEGAR approach for strict enforcement under the preferred semantics. Here we observe that Open-WBO solves most of the generated instances faster than LMHS. The

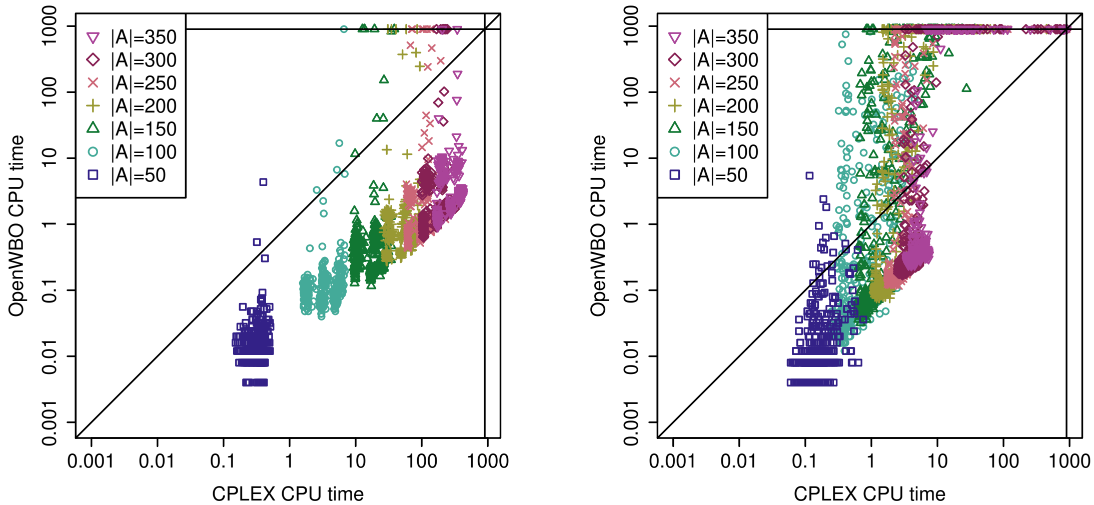


Figure 15: OpenWBO versus CPLEX on strict complete (left), non-strict stable (right)

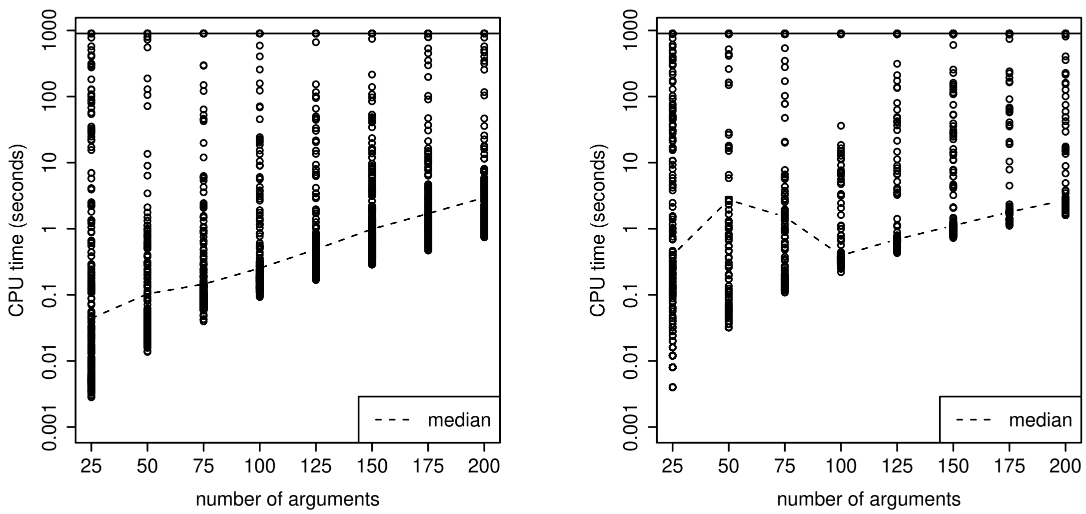


Figure 16: Σ_2^P-complete strict extension enforcement under the preferred (left) and semi-stable (right) semantics.

results of the empirical evaluation for stage semantics are not shown here, since the solvers time out on most of the instances even at 25 arguments. This is due to using the conflict-free semantics as the base abstraction, which basically just tells the MaxSAT solver to remove all attacks inside the enforced set. This results in the solver guessing truth assignments, i.e., the attack structure, without any further constraints. Therefore in practice the algorithm has too many iterations in the CEGAR loop, resulting in high runtimes. To improve the approach, we hypothesize that a more restricting base abstraction, or better refinement clauses, are needed.

In summary, we have shown that using MaxSAT solvers and a CEGAR-based approach for extension enforcement is practically viable, solving instances with 200 arguments and beyond on most of the problem variants. In general, core-guided MaxSAT solvers appear to provide the best alternative in terms of the median runtimes.

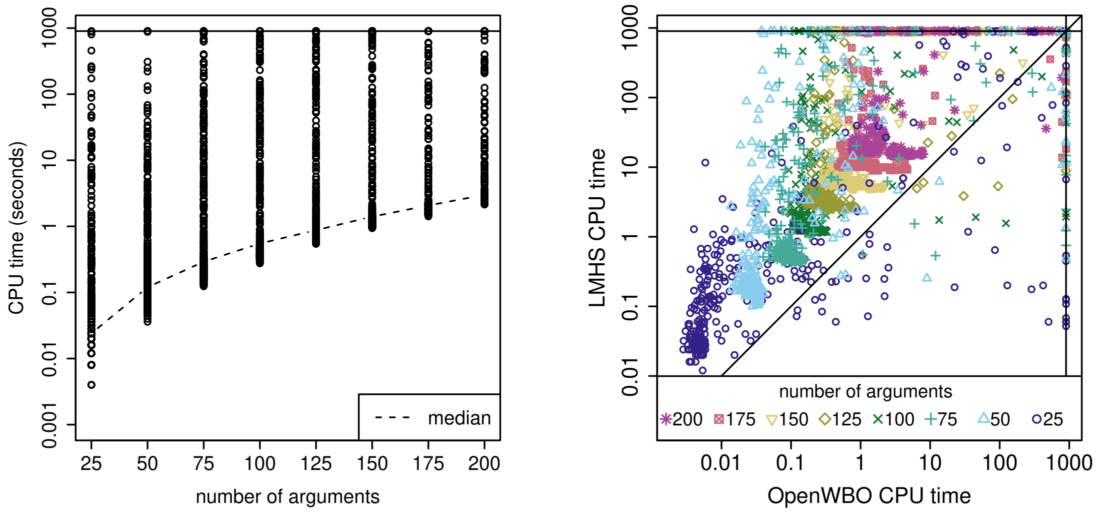


Figure 17: Left: Σ_2^P-complete non-strict extension enforcement under the semi-stable semantics; right: OpenWBO versus LMHS on strict preferred.

8. Other Extension Enforcement Variants

Finally, we shortly discuss other variants of extension enforcement in light of our approach as well as the related complexity questions. Baumann and Brewka (2010) consider enforcement under so-called expansions of an AF $F = (A, R)$, which result in AF $F' = (A \cup A', R \cup R')$ with new arguments A' and new attacks R' such that $A \cap A' = R \cap R' = \emptyset$ and one of A' or R' is non-empty. An expansion is normal if for each $(a, b) \in R'$ we have $a \in A'$ or $b \in A'$; strong if $a \notin A$; and weak if $b \notin A$. The tasks for enforcement under these variants are the same as for strict and non-strict, with the additional requirement that the enforcing AF is to be a normal, strong, or weak expansion. Analogously to the complexity results on argument-fixed enforcement already presented in this paper, we give complexity results for non-strict enforcement under normal, strong, or weak expansions for a *fixed* set A' of additional arguments, i.e., A' is part of the input. A proof is provided in Appendix A.

Theorem 11. *Non-strict enforcement under normal, strong, or weak expansions*

- *for admissible, complete, preferred, and stable semantics is* NP*-complete; and*
- *for semi-stable and stage semantics is* Σ_2^P*-complete.*

Argument-fixed extension enforcement, as considered in this work, allows arbitrary changes to the attack structure, and does not allow additional arguments. However, by adjusting the MaxSAT encoding in an appropriate way, further variants of these problems can be solved via the same approach, including capturing normal, strong, and weak expansions under a bounded number of additional arguments. First of all, as also pointed out by Coste-Marquis et al. (2015), enforcement under expansions can be encoded via additional hard constraints for a fixed set A' of additional arguments; the same holds for the MaxSAT encodings. In particular, our encodings allow for introducing any bounded number of additional arguments A' by considering $A \cup A'$ instead of the set A in the clauses of the encoding. However, we note that it is not clear under which condition the cost of optimal solutions is preserved when restricting these problems by considering a fixed set A'

(e.g., a singleton set). In fact, the following example shows that the costs of optimal solutions—and the sets of optimal solutions—do not in general coincide. Consider strictly enforcing $\{t_1, t_2\}$ under semi-stable semantics and weak expansions. The AF in Figure 18 requires two new arguments (for each argument t_1 and t_2 we need a new argument to extend their range); restricting A' to singleton sets does not yield any solutions.

More control on the attack structures of interest can be obtained by fixing a part of the attack structure of the solution AF which is accomplished by making the corresponding soft clauses hard. In other words, if we want the attacks $R^+ \subseteq A \times A$ to be present, and the attacks $R^- \subseteq A \times A$ not to be present in the solution AF, we can introduce the hard clauses

$$\bigwedge_{(a,b)\in R^+} r_{a,b} \wedge \bigwedge_{(a,b)\in R^-} \neg r_{a,b}$$

to the encoding, and modify the soft clauses by enumerating through the set $(A \times A) \setminus (R^+ \cup R^-)$ instead of $A \times A$. By further including a bounded number of additional arguments in the encodings, this allows for capturing normal, strong, and weak expansions by restricting changes to the attack structure according to the restrictions imposed by these forms of expansions. Furthermore, our CEGAR algorithm (Algorithm 1) with adapted hard constraints can also be applied for the preferred, semi-stable, and stage semantics under normal, strong, or weak expansions. However, as noted by Baumann and Brewka (2010) and Coste-Marquis et al. (2015), solutions to extension enforcement are not guaranteed to exist under normal, strong, or weak expansions.

Weighted argument systems, as defined by Dunne et al. (2011), are an extension of Dung's argumentation frameworks. In addition to the arguments and attacks, a non-zero weight is assigned to each attack, corresponding to the relative strength of the attack. Similarly to this approach, the enforcement problem can be modified by introducing a non-zero weight to each $(a, b) \in A \times A$ via the weight function $w \colon A \times A \to \mathbb{Z}_+$. Intuitively, if $(a, b) \in R$, the weight $w(a, b)$ corresponds to the strength of the existing attack, i.e., how reluctant we are to remove it, and if $(a, b) \notin R$, the weight $w(a, b)$ expresses how reluctant we are to add the attack. Now, using weighted partial MaxSAT instead of partial MaxSAT, this problem can be solved via the same approach by assigning to each soft clause $r'_{a,b}$ the weight $w(a, b)$.

9. Conclusions

The study of computational aspects of argumentation dynamics is a topical area of research in AI argumentation. We presented complexity results and novel algorithms, based on a declarative optimization approach, for several variants of argument-fixed extension enforcement. As the main contributions, on the theoretical side we presented a nearly complete computational complexity map of the considered problem variants. Complementing the theoretical analysis, we proposed algorithms for the variants, ranging from constructive polytime results to procedures going beyond NP for the second-level complete problem variants, and presented empirical results from an extended empirical evaluation of our system implementation of the algorithms on the scalability of

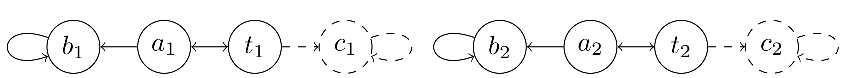


Figure 18: Enforcement under weak expansions

the approach. Further improving the efficiency of the approach both via understanding what makes enforcement instances hard and via employing optimization solvers incrementally with the CEGAR approach, as well as studying potential extensions of the algorithmic ideas presented in this article to other types of argumentation dynamics, are important aspects of future work. For example, extending the MaxSAT-based approach to cover extension enforcement under the grounded semantics is a nontrivial problem. Scalability of the presented algorithms on other argumentation framework generation models, such as those proposed by Cerutti, Giacomin, and Vallati (2016) as well as ones arising from possible application scenarios, would also be interesting to study.

Acknowledgments

The authors gratefully acknowledge financial support from the Academy of Finland under grants 251170 COIN Centre of Excellence in Computational Inference Research, 276412, and 284591; DoCS Doctoral School in Computer Science and Research Funds of the University of Helsinki; and the Austrian Science Fund (FWF): P30168-N31.

Appendix A. Proofs

Lemma 7. *Let X be a set, $choice(X) = F_g = (A_g, R_g)$, and $T = \{t_1, t_2, t_3\}$ with $T \subseteq A_g$. Further, let $F = (A, R)$ be an AF such that $A_g \subseteq A$, $R_g \subseteq R$, and $\nexists (a, a') \in R$ with $a' \in A_g$ and $a \in A \setminus A_g$. We define*

$$\hat{F} = \{F' \mid F' = (A, R'), T \subseteq \mathcal{F}_{F'}(T), T \in cf(F'),$$
$$\mathcal{F}_{F'}(T) \cap \{d_1^x, d_2^x \mid x \in X\} = \emptyset, |R \Delta R'| \leq 3 \cdot |X|\}.$$

For each $F' \in \hat{F}$ with $F' = (A, R')$ and each $x \in X$ exactly one of the following statements holds.

1. *There exists a $(t, x) \in R'$ with $t \in T$, and there does not exist an $(\overline{x}, t') \in R'$ with $t' \in T$.*
2. *There exists a $(t, \overline{x}) \in R'$ with $t \in T$, and there does not exist an $(x, t') \in R'$ with $t' \in T$.*

Further, it holds that $(R \Delta R' \subseteq (T \cup X \cup \overline{X}) \times (T \cup X \cup \overline{X}))$.

Proof. Let $\hat{F}$ be defined as in the lemma and let $F' \in \hat{F}$. For convenience we use the shorthand $D_x = \{d_1^x, d_2^x\}$. Consider

$$R_x = \{(a, b) \mid a \in A, b \in \{x, \overline{x}\} \cup D_x\} \cup \{(z, t) \mid z \in \{x, \overline{x}\} \cup D_x, t \in T\}.$$

Here R_x contains all attacks onto arguments in the "gadget" made for x and all attacks from these gadget arguments to T. It holds that for each $x, x' \in X$ with $x \neq x'$ that $R_x \cap R_{x'} = \emptyset$. We now show that for each $x \in X$ it holds that $|(R \cap R_x) \Delta (R' \cap R_x)| = 3$. Let $x \in X$ and $m = |(R \cap R_x) \Delta (R' \cap R_x)|$. We consider each case separately.

- Case $m = 0$: Then $T \not\subseteq \mathcal{F}_{F'}(T)$, since $(x, t_1) \in R'$ and $\nexists (t, x) \in R'$ for $t \in T$.
- case $m = 1$: Then $T \not\subseteq \mathcal{F}_{F'}(T)$, since both $\{x, \overline{x}\}$ attack T in F via four attacks. Removal of one of these four attacks results in T not defending itself in F'; likewise adding an attack from T onto $\{x, \overline{x}\}$ cannot defend T against both arguments x and $\overline{x}$.

- Case $m = 2$: We consider here two subcases for F', namely that both $\{x, \overline{x}\}$ are attacked by T, or at least one of these arguments is not attacked by T.
  - Assume it holds that $\{(t, x), (t', \overline{x})\} \subseteq R'$ for some $t, t' \in T$. This implies that $D_x \subseteq \mathcal{F}_{F'}(T)$, since all attackers of D_x are attacked by T (see also Figure 6b).
  - Assume there exists a $z \in \{x, \overline{x}\}$ such that there is no $t \in T$ with $(t, z) \in R'$. Consider again two subcases. Either there exists an attack from z to T or not in F'. If such an attack does not exist, then there must exist an undefended attack from $\{x, \overline{x}\} \setminus \{z\}$ onto T (removing both attacks from z to T are the only changes on attacks in R_x). If there is an attack from z to T in F', then T does not defend itself against this attack.
- Case $m = 3$: This case is possible, as witnessed by, e.g., one of the two statements in the lemma.
- Case $m > 3$: Since $|R \Delta R'| \leq 3 \cdot |X|$, it holds that if $m > 3$, then there exists an $x' \in X$ such that $|(R \cap R_{x'} \Delta R' \cap R_{x'})| < 3$ which is not possible (see previous cases).

Thus, under the assumptions of the lemma, for each $x \in X$ we have $|((R \cap R_x) \Delta (R' \cap R_x))| = 3$.

We now show that exactly one of the statements in the lemma holds. We use statement 1 and statement 2 as shorthands for the two statements. Consider again $x \in X$ and R_x as defined above. If both statement 1 and statement 2 hold for $x \in X$, then $|(R \cap R_x \Delta R' \cap R_x)| > 3$. This is a contradiction. Now suppose both statements do not hold. For convenience, we state both again in negated form.

$$\forall t \in T \text{ it holds that } (t, x) \notin R' \tag{1}$$
$$\exists t' \in T \text{ s.t. } (\overline{x}, t') \in R' \tag{2}$$
$$\forall t \in T \text{ it holds that } (t, \overline{x}) \notin R' \tag{3}$$
$$\exists t' \in T \text{ s.t. } (x, t') \in R' \tag{4}$$

Negated statement 1 is (1) or (2) and negated statement 2 is (3) or (4).

We again proceed by case analysis. We consider the four cases where both negated statements hold (both negated statements consist of a disjunction of two assertions).

- Case (1) and (3) hold. In this case T does not attack $\{x, \overline{x}\}$. Since in F there are four attacks from $\{x, \overline{x}\}$ onto T at least one of them must be present in F'. Therefore T does not defend itself.
- Case (1) and (4) hold. Here T does not attack x and x attacks T. Thus T does not defend itself.
- Case (2) and (3) hold. Symmetric to the second case ((1) and (4) hold).
- Case (2) and (4) hold. In this case both $\{x, \overline{x}\}$ attack T. To defend T in F' against these two arguments, two attacks from T onto both x and $\overline{x}$ are required to be present in F'. We further know that $D_x \cap \mathcal{F}_{F'}(T) = \emptyset$. Since T defends both arguments in D_x from potential attacks from $\{x, \overline{x}\}$, we can infer that both d_1^x and d_2^x are attacked by at least one argument in $A \setminus \{x, \overline{x}\}$. This is a contradiction, since in F there are no such attacks and one is required to be present in F' for each argument in D_x, which would imply that $|((R \cap R_x) \Delta (R' \cap R_x))| > 3$ (two changes for attacks from T onto $\{x, \overline{x}\}$ and two attacks onto D_x).

Therefore either statement 1 holds or statement 2 holds but not both. This also implies the final claim in the lemma, i.e., $R\Delta R' \subseteq (T \cup X \cup \overline{X} \times T \cup X \cup \overline{X})$. □

Theorem 9. *Strict extension enforcement for preferred and semi-stable semantics is Σ_2^P-complete.*

Proof. Membership follows from guessing a new attack structure for a given enforcement problem, and verifying with an NP-oracle whether the given set is a preferred, resp. semi-stable, extension (Dunne & Caminada, 2008; Caminada et al., 2012; Dvořák & Woltran, 2010). Further, we check whether the symmetric difference between the new and original attack structure does not exceed the given bound.

For hardness we reduce from the problem of checking whether a given closed quantified Boolean formula (QBF) in prenex normal form $\psi = \exists X \forall Y \phi$ is satisfiable (or true). W.l.o.g. let ϕ be in disjunctive normal form. We construct the following instance of our enforcement problem with $(A_g, R_g) = \mathit{choice}(X)$ and C the set of conjunctions of ϕ. Within this proof, we view conjunctions also as sets. Let $V = \{v_x \mid x \in X\}$.

$$
\begin{aligned}
A =& A_g \cup V \cup \overline{V} \cup Y \cup \overline{Y} \cup C \cup \{\overline{f}, f\}\\
R =& R_g \cup \{(y, \overline{y}), (\overline{y}, y) \mid y \in Y\} \cup\\
& \{(v_x, \overline{v_x}), (\overline{v_x}, v_x) \mid x \in X\} \cup\\
& \{(x, v_x) \mid x \in X\} \cup \{(\overline{x}, \overline{v_x}) \mid \overline{x} \in \overline{X}\} \cup\\
& \{(v_x, c) \mid v_x \in V, \neg x \in c\} \cup\\
& \{(\overline{v_x}, c) \mid \overline{v_x} \in \overline{V}, x \in c\} \cup\\
& \{(y, c) \mid y \in Y, \neg y \in c\} \cup\\
& \{(\overline{y}, c) \mid \overline{y} \in \overline{Y}, y \in c\} \cup\\
& \{(c, \overline{f} \mid c \in C\} \cup \{(f, f)\} \cup\\
& \{(f, z) \mid z \in V \cup \overline{V} \cup Y \cup \overline{Y}\}
\end{aligned}
$$

Finally, we set $k = 3 \cdot |X|$. The reduction is shown in Figure 19 and can be computed in polynomial time. Let $\sigma \in \{\mathit{prf}, \mathit{sem}\}$. We now claim that there exists an $F' = (A, R')$ such that $T \in \sigma(F')$ with $|R\Delta R'| \leq 3 \cdot |X|$ iff ψ is satisfiable. For some truth assignment τ defined on variables $X \cup Y$ we denote by $\tau|_X$ the restriction of τ to X, i.e., the truth assignment τ' that is defined on X and $\tau(x) = \tau'(x)$ for all $x \in X$.

Assume that ψ is satisfiable. This means there is a partial truth assignment τ defined on variables in X such that for every total interpretation τ' on $X \cup Y$ with $\tau'|_X = \tau$ we have $\tau' \models \phi$. We define two sets of attacks D_1 and D_2 as follows.

$$
\begin{aligned}
D_1 =& \{(x, t) \mid x \in X, \tau(x) = 0, t \in \{t_1, t_2\}\} \cup\\
& \{(\overline{x}, t) \mid x \in X, \tau(x) = 1, t \in \{t_1, t_2\}\},\\
D_2 =& \{(t_3, x) \mid x \in X, \tau(x) = 1\} \cup\\
& \{(t_3, \overline{x}) \mid x \in X, \tau(x) = 0\}.
\end{aligned}
$$

We now define a new AF $F' = (A, R')$ with $R' = (R \setminus D_1) \cup D_2$. It holds that $|R\Delta R'| = 3 \cdot |X|$. We now claim that $T \in \mathit{sem}(F')$ (which implies that $T \in \mathit{prf}(F')$).

It holds that $T \in grd(F')$ because t_3 is not attacked at all in F' and for each $x \in X$ there are two attacks from $\{x, \overline{x}\}$ onto T which are counterattacked by t_3, thus $\{t_1, t_2\}$ is defended by t_3.

We now show that there is no admissible set T' such that $T \subset T'$. This claim implies that T is a semi-stable extension of F'. Suppose the contrary, i.e., $\exists T' \in adm(F')$ for which $T \subset T'$. First some basic observations. No self-attacking argument in F' can be in T'. For all $x \in X$ it holds that d_1^x and d_2^x are not in T', since one of their attackers (x or $\overline{x}$) is not attacked by any argument except themselves. Therefore, the only candidate arguments in $A \setminus T$ to be present in T' are $\overline{f}$ and the arguments in the sets Y, $\overline{Y}$, V, and $\overline{V}$. A further crucial observation is that if T' contains any argument from the sets Y, $\overline{Y}$, V, or $\overline{V}$, then also $\overline{f}$ must be in T' (only $\overline{f}$ defends these arguments against f). Thus, T' must contain $\overline{f}$. Since $\overline{f}$ is attacked by all $c \in C$, at least one attacker of each c must be present in T'. From these arguments we create a partial truth assignment as follows.

$$\tau''(z) = \begin{cases} 1 & \text{if } v_z \in V \cap T' \\ 1 & \text{if } z \in Y \cap T' \\ 0 & \text{if } \overline{v_z} \in \overline{V} \cap T' \\ 0 & \text{if } \overline{z} \in \overline{Y} \cap T' \end{cases}$$

This truth assignment is well-defined, i.e., no variable is assigned both 1 and 0, since the corresponding arguments for assigning 1 resp. 0 are mutually attacking in F'. It holds that for each $x \in X$ if $\tau''(x)$ is defined on x, then $\tau''(x) = \tau(x)$. Suppose the contrary is true, i.e., $\tau''(x) \neq \tau(x)$. We consider the case that $\tau''(x) = 1$ ($\tau''(x) = 0$ is symmetric). Then $v_x \in T'$ and there is no attack from T on x that is in turn attacking v_x. This is in contradiction with T' being admissible.

We now prove that $\tau'' \not\models \phi$ (in particular the partial truth assignment suffices to show this even for all completions to all variables in ϕ). First note that $\phi[\tau'']$ is unsatisfiable iff $\forall c \in C, c[\tau'']$ is unsatisfiable. The latter statement holds if $\forall c \in C, \exists l \in c, \tau''(l) = 0$.

$$\begin{aligned}
&\forall c \in C, \exists l \in c, \tau''(l) = 0 \\
\text{iff } &\forall c \in C, \exists l \in c \\
&\qquad \text{if } l = z \text{ then } \tau''(z) = 0 \text{ and} \\
&\qquad \text{if } l = \neg z \text{ then } \tau''(z) = 1 \\
\text{iff } &\forall c \in C, \exists l \in c \\
&\qquad \text{if } l = x, x \in X \text{ then } \overline{v_x} \in T' \text{ and} \\
&\qquad \text{if } l = y, y \in Y \text{ then } \overline{y} \in T' \text{ and} \\
&\qquad \text{if } l = \neg x, x \in X \text{ then } v_x \in T' \text{ and} \\
&\qquad \text{if } l = \neg y, y \in Y \text{ then } y \in T' \\
\text{iff } &\forall c \in C, \exists (z, c) \in R' \\
&\qquad \text{if } z = \overline{v_x}, x \in X \text{ then } \overline{v_x} \in T' \text{ and} \\
&\qquad \text{if } z = \overline{y}, y \in Y \text{ then } \overline{y} \in T' \text{ and} \\
&\qquad \text{if } z = v_x, x \in X \text{ then } v_x \in T' \text{ and} \\
&\qquad \text{if } z = y, y \in Y \text{ then } y \in T' \\
\text{iff } &\forall c \in C, \exists (z, c) \in R' \text{ it holds that } z \in T' \\
\text{iff } &\overline{f} \in \mathcal{F}_{F'}(T').
\end{aligned}$$

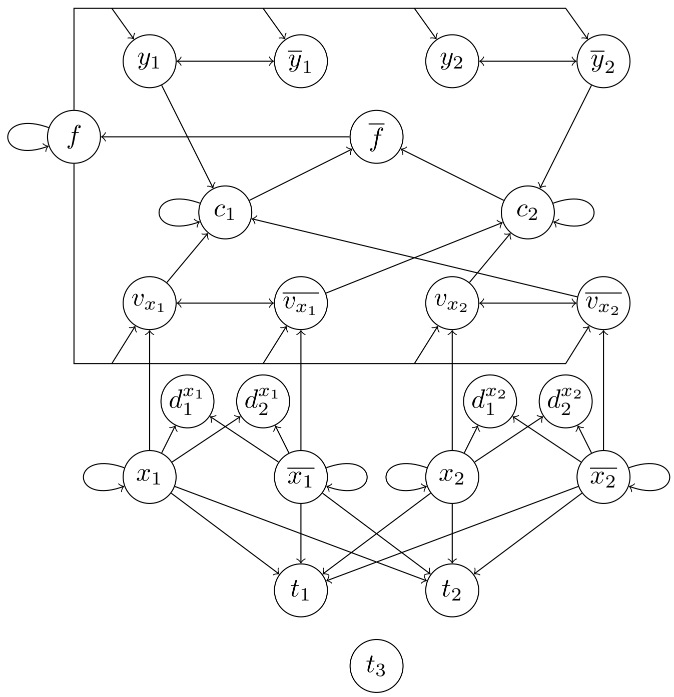


Figure 19: Reduction for Theorem 9 with clauses $c_1 = (\neg x_1 \wedge x_2 \wedge \neg y_1)$ and $c_2 = (x_1 \wedge \neg x_2 \wedge y_2)$.

Thus $\tau'' \not\models \phi$ (or if τ'' is partial, then every completion of τ'' to all variables does not satisfy ϕ).

The previous observations imply that there exists an interpretation τ' defined on $X \cup Y$ such that $\tau'|_X = \tau$ with $\tau' \not\models \phi$. We define τ' by assigning the same truth values as τ to the variables in X and the same truth values to the variables in Y as τ''. It follows that τ' is not a model of ϕ. Therefore the assumption that $\overline{f} \in T'$ leads to a contradiction. This in turn implies that there is no strictly larger admissible set than T containing $\overline{f}$. This means that there is no strictly larger admissible set than T in F' and also that $T \in sem(F')$ (in fact, T is the only complete extension of F'). If T is semi-stable in F', then T is also preferred in F'.

We now proceed to the other direction. Assume that there exists an $F' = (A, R')$ such that $T \in \sigma(F')$ and $|R \Delta R'| \leq 3 \cdot |X|$. Due to Lemma 7 we can infer that $R \Delta R'$ contains only attacks from T to $X \cup \overline{X}$ as specified in that lemma. We construct a partial interpretation τ on X by

$$\tau(x) = \begin{cases} 1 & \text{if } \exists (t, x) \in R', t \in T \\ 0 & \text{if } \exists (t, \overline{x}) \in R', t \in T. \end{cases}$$

We now claim that for any interpretation τ' defined on $X \cup Y$ such that $\tau'|_X = \tau$ we have $\tau' \models \phi$. Suppose the contrary, that is, there exists such a τ' such that $\tau' \not\models \phi$. Then for all $c \in C$ we have

$\tau' \not\models c$. We now construct a set of arguments T' as follows.

$$\begin{aligned}T' = T \cup \{\overline{f}\} \cup \\ \{v_x \mid v_x \in V, \tau'(x) = 1\} \cup \\ \{\overline{v_x} \mid \overline{v_x} \in \overline{V}, \tau'(x) = 0\} \cup \\ \{y \mid y \in Y, \tau'(y) = 1\} \cup \\ \{\overline{y} \mid \overline{y} \in \overline{Y}, \tau'(y) = 0\}\end{aligned}$$

We claim that T' is a semi-stable extension of F'. It is immediate that T' defends itself against all attacks in F' except for those coming from C. Let $(c, \overline{f})$ be such an attack for $c \in C$. By assumption it holds that there exists a literal l in c such that $\tau' \not\models l$. Then the variable of l is assigned the complementary value in τ'. The corresponding argument (v_x or $\overline{v_x}$ if variable is in X, otherwise y or $\overline{y}$ if variable is in Y) is present in T', thus defending $\overline{f}$ against c. This holds for all $c \in C$. Thus T' is admissible. It is immediate that T' is also semi-stable (every argument except d_1^x, d_2^x and one of $x, \overline{x}$ for each $x \in X$ is in the range of T'; these arguments are never in the range of an admissible set). Since $T \subset T'$ it holds that $T \notin \mathit{prf}(F')$ and in turn $T \notin \mathit{sem}(F')$. This is a contradiction. Thus ψ is satisfiable. $\square$

Theorem 11. *Non-strict enforcement under normal, strong, or weak expansions*

- *for admissible, complete, preferred, and stable semantics is* NP*-complete; and*
- *for semi-stable and stage semantics is* Σ_2^P*-complete.*

Proof. First note that non-strict extension enforcement coincides for admissible, complete, and preferred semantics when we consider one type of expansions. In other words, non-strict extension enforcement coincides under these three semantics under normal expansions, and, as well under these three semantics, coincides under strong, and coincides under weak expansions. To see this, assume a set X is non-strictly enforced under admissible semantics via an expansion with the solution AF F. It holds that X is part of an admissible set of F, which implies that X is contained in a complete extension of F, and that X is contained in a preferred extension of F. Assume that X is non-strictly enforced under preferred semantics via an expansion with the solution AF F'. Then X is contained in a preferred extension E of F'. By definition, it holds that E is both complete and admissible, and, in turn, X is contained in a complete extension (admissible set) of F'.

Let $\sigma \in \{\mathit{adm}, \mathit{stb}, \mathit{sem}, \mathit{stg}\}$. Membership for all stated complexity results holds via a straightforward guess and check; guess a modified attack structure and superset of the given set to enforce, and check whether the superset is a σ-extension. One can verify in polynomial time whether a set is admissible or stable. The problem of verifying whether a set is a semi-stable or stage extension is in coNP.

For hardness, similarly as for non-strict extension enforcement with a fixed set of arguments (recall proofs of Theorem 3 and Theorem 4), we reduce from the problem of credulous acceptance of a given argument a in a given AF $F = (A, R)$ under semantics σ, i.e., checking whether a σ-extension $E \subseteq A$ exists with $a \in E$. The problem of credulous acceptance is NP-complete for admissible and stable semantics and Σ_2^P-complete for semi-stable and stage semantics. We reduce the problem of credulous acceptance to non-strict extension enforcement under expansion as follows. Construct a new AF $F' = (A', R)$ with $A' = A \cup \{b\}$ with $b \notin A$. The AF F' is a normal, strong, and weak expansion of F (no attacks added). From the definitions of the semantics it follows that for any $E \subseteq A$ it holds that $E \in \sigma(F)$ iff $(E \cup \{b\}) \in \sigma(F')$. For $\sigma' \in \{stb, sem, stg\}$ we have $b \in E$ for all $E \in \sigma'(F')$ (there are no attacks onto or from b in F'). For admissible semantics, we can infer that $E' \in adm(F')$ iff $E' \cup \{b\} \in adm(F')$ (again by the fact that b is neither attacked or attacks arguments in A'). Thus, if $E \in \sigma(F')$, then $E \cup \{b\} \in \sigma(F')$. This implies that an argument a is credulously accepted in F iff argument a is credulously accepted in F' iff one can non-strictly enforce $\{a\}$ under normal, strong, or weak expansions in F' with 0 modifications. $\square$

References

Alviano, M., Dodaro, C., & Ricca, F. (2015). A MaxSAT algorithm using cardinality constraints of bounded size. In Yang, Q., & Wooldridge, M. (Eds.), *Proc. IJCAI*, pp. 2677–2683. AAAI Press / IJCAI.

Amgoud, L., & Prade, H. (2009). Using arguments for making and explaining decisions. *Artif. Intell.*, *173*(3-4), 413–436.

Ansótegui, C., Didier, F., & Gabàs, J. (2015). Exploiting the structure of unsatisfiable cores in MaxSAT. In Yang, Q., & Wooldridge, M. (Eds.), *Proc. IJCAI*, pp. 283–289. AAAI Press.

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

Baumann, R. (2012a). Normal and strong expansion equivalence for argumentation frameworks. *Artif. Intell.*, *193*, 18–44.

Baumann, R. (2012b). What does it take to enforce an argument? Minimal change in abstract argumentation. In Raedt, L. D., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., & Lucas, P. J. F. (Eds.), *Proc. ECAI*, Vol. 242 of *Frontiers in Artificial Intelligence and Applications*, pp. 127–132. IOS Press.

Baumann, R., & Brewka, G. (2010). Expanding argumentation frameworks: Enforcing and monotonicity results. In Baroni, P., Cerutti, F., Giacomin, M., & Simari, G. R. (Eds.), *Proc. COMMA*, Vol. 216 of *Frontiers in Artificial Intelligence and Applications*, pp. 75–86. IOS Press.

Baumann, R., & Brewka, G. (2015). AGM meets abstract argumentation: Expansion and revision for Dung frameworks. In Yang, Q., & Wooldridge, M. (Eds.), *Proc. IJCAI*, pp. 2734–2740. AAAI Press.

Baumeister, D., Neugebauer, D., & Rothe, J. (2015a). Verification in attack-incomplete argumentation frameworks. In Walsh, T. (Ed.), *Proc. ADT*, Vol. 9346 of *Lecture Notes in Computer Science*, pp. 341–358. Springer.

Baumeister, D., Rothe, J., & Schadrack, H. (2015b). Verification in argument-incomplete argumentation frameworks. In Walsh, T. (Ed.), *Proc. ADT*, Vol. 9346 of *Lecture Notes in Computer Science*, pp. 359–376. Springer.

Bench-Capon, T. J. M., & Dunne, P. E. (2007). Argumentation in artificial intelligence. *Artif. Intell.*, *171*(10-15), 619–641.

Bench-Capon, T. J. M., Prakken, H., & Sartor, G. (2009). Argumentation in legal reasoning. In Simari, G., & Rahwan, I. (Eds.), *Argumentation in Artificial Intelligence*, pp. 363–382. Springer.

Besnard, P., & Doutre, S. (2004). Checking the acceptability of a set of arguments. In Delgrande, J. P., & Schaub, T. (Eds.), *Proc. NMR*, pp. 59–64.

Bisquert, P., Cayrol, C., de Saint-Cyr, F. D., & Lagasquie-Schiex, M. (2013). Enforcement in argumentation is a kind of update. In Liu, W., Subrahmanian, V. S., & Wijsen, J. (Eds.), *Proc. SUM*, Vol. 8078 of *Lecture Notes in Computer Science*, pp. 30–43. Springer.

Booth, R., Kaci, S., Rienstra, T., & van der Torre, L. W. N. (2013). A logical theory about dynamics in abstract argumentation. In Liu, W., Subrahmanian, V. S., & Wijsen, J. (Eds.), *Proc. SUM*, Vol. 8078 of *Lecture Notes in Computer Science*, pp. 148–161. Springer.

Caminada, M. W. A., Carnielli, W. A., & Dunne, P. E. (2012). Semi-stable semantics. *J. Logic Comput.*, *22*(5), 1207–1254.

Cayrol, C., de Saint-Cyr, F. D., & Lagasquie-Schiex, M. (2010). Change in abstract argumentation frameworks: Adding an argument. *J. Artif. Intell. Res.*, *38*, 49–84.

Cerutti, F., Dunne, P. E., Giacomin, M., & Vallati, M. (2014a). Computing preferred extensions in abstract argumentation: A SAT-based approach. In Black, E., Modgil, S., & Oren, N. (Eds.), *TAFA 2013 Revised Selected Papers*, Vol. 8306 of *Lecture Notes in Computer Science*, pp. 176–193. Springer.

Cerutti, F., Giacomin, M., & Vallati, M. (2014b). ArgSemSAT: Solving argumentation problems using SAT. In Parsons, S., Oren, N., Reed, C., & Cerutti, F. (Eds.), *Proc. COMMA*, Vol. 266 of *Frontiers in Artificial Intelligence and Applications*, pp. 455–456. IOS Press.

Cerutti, F., Giacomin, M., & Vallati, M. (2016). Generating structured argumentation frameworks: Afbenchgen2. In Baroni, P., Gordon, T. F., Scheffler, T., & Stede, M. (Eds.), *Proc. COMMA*, Vol. 287 of *Frontiers in Artificial Intelligence and Applications*, pp. 467–468. IOS Press.

Clarke, E. M., Grumberg, O., Jha, S., Lu, Y., & Veith, H. (2003). Counterexample-guided abstraction refinement for symbolic model checking. *J. ACM*, *50*(5), 752–794.

Clarke, E. M., Gupta, A., & Strichman, O. (2004). SAT-based counterexample-guided abstraction refinement. *IEEE TCAD*, *23*(7), 1113–1123.

Coste-Marquis, S., Devred, C., & Marquis, P. (2005). Symmetric argumentation frameworks. In Godo, L. (Ed.), *Proc. ECSQARU*, Vol. 3571 of *Lecture Notes in Computer Science*, pp. 317–328. Springer.

Coste-Marquis, S., Konieczny, S., Mailly, J., & Marquis, P. (2014a). On the revision of argumentation systems: Minimal change of arguments statuses. In Baral, C., Giacomo, G. D., & Eiter, T. (Eds.), *Proc. KR*, pp. 52–61. AAAI Press.

Coste-Marquis, S., Konieczny, S., Mailly, J., & Marquis, P. (2014b). A translation-based approach for revision of argumentation frameworks. In Fermé, E., & Leite, J. (Eds.), *Proc. JELIA*, Vol. 8761 of *Lecture Notes in Computer Science*, pp. 397–411. Springer.

Coste-Marquis, S., Konieczny, S., Mailly, J., & Marquis, P. (2015). Extension enforcement in abstract argumentation as an optimization problem. In Yang, Q., & Wooldridge, M. (Eds.), *Proc. IJCAI*, pp. 2876–2882. AAAI Press.

Davies, J., & Bacchus, F. (2013). Exploiting the power of MIP solvers in MAXSAT. In Järvisalo, M., & Gelder, A. V. (Eds.), *Proc. SAT*, Vol. 7962 of *Lecture Notes in Computer Science*, pp. 166–181. Springer.

de Saint-Cyr, F. D., Bisquert, P., Cayrol, C., & Lagasquie-Schiex, M. (2016). Argumentation update in YALLA (Yet Another Logic Language for Argumentation). *Int. J. Approx. Reasoning*, *75*, 57–92.

Delobelle, J., Haret, A., Konieczny, S., Mailly, J., Rossit, J., & Woltran, S. (2016). Merging of abstract argumentation frameworks. In Baral, C., Delgrande, J. P., & Wolter, F. (Eds.), *Proc. KR*, pp. 33–42. AAAI Press.

Delobelle, J., Konieczny, S., & Vesic, S. (2015). On the aggregation of argumentation frameworks. In Yang, Q., & Wooldridge, M. (Eds.), *Proc. IJCAI*, pp. 2911–2917. AAAI Press.

Diller, M., Haret, A., Linsbichler, T., Rümmele, S., & Woltran, S. (2015). An extension-based approach to belief revision in abstract argumentation. In Yang, Q., & Wooldridge, M. (Eds.), *Proc. IJCAI*, pp. 2926–2932. AAAI Press.

Dimopoulos, Y., & Torres, A. (1996). Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, *170*(1-2), 209–244.

Donaldson, A. F., Kaiser, A., Kroening, D., Tautschnig, M., & Wahl, T. (2012). Counterexample-guided abstraction refinement for symmetric concurrent programs. *Formal Methods in System Design*, *41*(1), 25–44.

Doutre, S., Herzig, A., & Perrussel, L. (2014). A dynamic logic framework for abstract argumentation. In Baral, C., Giacomo, G. D., & Eiter, T. (Eds.), *Proc. KR*, pp. 62–71. AAAI Press.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, *77*(2), 321–358.

Dunne, P. E., & Caminada, M. (2008). Computational complexity of semi-stable semantics in abstract argumentation frameworks. In Hölldobler, S., Lutz, C., & Wansing, H. (Eds.), *Proc. JELIA*, Vol. 5293 of *Lecture Notes in Computer Science*, pp. 153–165. Springer.

Dunne, P. E., Hunter, A., McBurney, P., Parsons, S., & Wooldridge, M. (2011). Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artif. Intell.*, *175*(2), 457–486.

Dunne, P. E., & Wooldridge, M. (2009). Complexity of abstract argumentation. In Simari, G., & Rahwan, I. (Eds.), *Argumentation in Artificial Intelligence*, pp. 85–104. Springer.

Dvořák, W., Järvisalo, M., Wallner, J. P., & Woltran, S. (2014). Complexity-sensitive decision procedures for abstract argumentation. *Artif. Intell.*, *206*, 53–78.

Dvořák, W., & Woltran, S. (2010). Complexity of semi-stable and stage semantics in argumentation frameworks. *Inform. Process. Lett.*, *110*(11), 425–430.

Egly, U., Gaggl, S. A., & Woltran, S. (2010). Answer-set programming encodings for argumentation frameworks. *Argument and Computation*, *1*(2), 147–177.

Endriss, U., & Grandi, U. (2017). Graph aggregation. *Artif. Intell.*, *245*, 86–114.

Janota, M., Grigore, R., & Marques-Silva, J. (2010). Counterexample guided abstraction refinement algorithm for propositional circumscription. In Janhunen, T., & Niemelä, I. (Eds.), *Proc. JELIA*, Vol. 6341 of *Lecture Notes in Computer Science*, pp. 195–207. Springer.

Maher, M. J. (2016a). Corrupt strategic argumentation: The ideal and the naive. In Kang, B. H., & Bai, Q. (Eds.), *Proc. AUS-AI*, Vol. 9992 of *Lecture Notes in Computer Science*, pp. 17–28. Springer.

Maher, M. J. (2016b). Resistance to corruption of general strategic argumentation. In Baldoni, M., Chopra, A. K., Son, T. C., Hirayama, K., & Torroni, P. (Eds.), *Proc. PRIMA*, Vol. 9862 of *Lecture Notes in Computer Science*, pp. 61–75. Springer.

Maher, M. J. (2016c). Resistance to corruption of strategic argumentation. In Schuurmans, D., & Wellman, M. P. (Eds.), *Proc. AAAI*, pp. 1030–1036. AAAI Press.

Mailly, J. (2016). Using enthymemes to fill the gap between logical argumentation and revision of abstract argumentation frameworks. In Kern-Isberner, G., & Wassermann, R. (Eds.), *Proc. NMR*, Vol. 852 of *Technical Reports in Computer Science*, pp. 95–104. Technische Universität Dortmund.

Martins, R., Manquinho, V. M., & Lynce, I. (2014). Open-WBO: A modular MaxSAT solver. In Sinz, C., & Egly, U. (Eds.), *Proc. SAT*, Vol. 8561 of *Lecture Notes in Computer Science*, pp. 438–445. Springer.

McBurney, P., Parsons, S., & Rahwan, I. (Eds.). (2012). *ArgMAS 2011 Revised Selected Papers*, Vol. 7543 of *Lecture Notes in Computer Science*. Springer.

Morgado, A., Ignatiev, A., & Marques-Silva, J. (2015). MSCG: Robust core-guided MaxSAT solving. *Journal on Satisfiability, Boolean Modeling and Computation*, *9*, 129–134.

Niskanen, A., Wallner, J. P., & Järvisalo, M. (2016). Pakota: A system for enforcement in abstract argumentation. In Michael, L., & Kakas, A. C. (Eds.), *Proc. JELIA*, Vol. 10021 of *Lecture Notes in Computer Science*, pp. 385–400. Springer.

Nofal, S., Atkinson, K., & Dunne, P. E. (2014). Algorithms for decision problems in argument systems under preferred semantics. *Artif. Intell.*, *207*, 23–51.

Papadimitriou, C. H. (1994). *Computational complexity*. Addison-Wesley.

Saikko, P., Berg, J., & Järvisalo, M. (2016). LMHS: A SAT-IP hybrid MaxSAT solver. In Creignou, N., & Berre, D. L. (Eds.), *Proc. SAT*, Vol. 9710 of *Lecture Notes in Computer Science*, pp. 539–546. Springer.

Wallner, J. P., Niskanen, A., & Järvisalo, M. (2016). Complexity results and algorithms for extension enforcement in abstract argumentation. In Schuurmans, D., & Wellman, M. (Eds.), *Proc. AAAI*, pp. 1088–1094. AAAI Press.