Argumentation Frameworks Induced by Assumption-based Argumentation: Relating Size and Complexity

Tuomo Lehtonen[1] , **Anna Rapberger**[2] , **Markus Ulbricht**[3] , **Johannes P. Wallner**[4]
[1]University of Helsinki, Department of Computer Science
[2]TU Wien, Institute of Logic and Computation
[3]Leipzig University, Department of Computer Science
[4]Graz University of Technology, Institute of Software Technology

Abstract

A key ingredient of computational argumentation in AI is the generation of arguments in favor of or against claims under scrutiny. In this paper we look at the complexity of argument construction and reasoning in the prominent structured formalism of assumption-based argumentation (ABA). We point out that reasoning in ABA by means of constructing an abstract argumentation framework (AF) gives rise to two main sources of complexity: (i) constructing the AF and (ii) reasoning within the constructed graph. Since both steps are intractable in general, it is no surprise that the best performing state-of-the-art ABA reasoners skip the instantiation procedure entirely and perform tasks directly on the input knowledge base. Driven by this observation, we identify and study atomic and symmetric ABA, two ABA fragments that preserve the expressive power of general ABA, and that can be utilized to have milder complexity in the first or second step. We show that using atomic ABA allows for an instantiation procedure for general ABA leading to polynomially-bounded AFs and that symmetric ABA can be used to create AFs that have mild complexity to reason on. By an experimental evaluation, we show that using the former approach with modern AF solvers can be competitive with state-of-the-art ABA solvers, improving on previous AF instantiation approaches that are hindered by intractable argument construction.

1 Introduction

Computational models of argumentation are a central approach within non-monotonic reasoning (Baroni et al. 2018) with a variety of applications (Atkinson et al. 2017) in, e.g., legal or medical reasoning. Key to many approaches to computational argumentation are so-called structured argumentation formalisms which specify a formal argumentative workflow. Among the prominent approaches in the field are assumption-based argumentation (ABA) (Bondarenko et al. 1997; Čyras et al. 2018), ASPIC$^+$ (Modgil and Prakken 2013), defeasible logic programming (DeLP) (García and Simari 2004), and deductive argumentation (Besnard and Hunter 2008). Reasoning within these formalisms is often carried out by instantiating argument structures and conflicts among these arguments from (rule-based) knowledge bases in a principled manner. The resulting arguments and conflicts are referred to as abstract argumentation frameworks (AFs) (Dung 1995). Argumentation semantics then define argumentative acceptability on an AF such that conclusions can be drawn for the original knowledge base.

In the present paper, we focus on ABA which is well studied and has applications in, e.g., decision making (Craven et al. 2012; Čyras et al. 2021; Fan et al. 2014). In an ABA framework (ABAF) argumentative reasoning can be carried out by instantiating arguments as derivations in the given rule base and attacks between arguments based on contraries among the derivations.

There are multiple reasons for studying the AF obtained from a given knowledge base. From a technical point of view, there is an abundance of research concerned with AFs which can be applied to assess the instantiated AF, see, e.g., the Handbook of Formal Argumentation (Baroni et al. 2018) for an overview. Thus, many typical research questions can be answered out of the box after converting the knowledge base. Moreover, since AFs are directed graphs, they are accessible and user-friendly; information on the relations between arguments that is implicit in the knowledge base is made explicit and clear within the graphical framework.

Nevertheless, the instantiation procedure comes with computational costs and consequently state-of-the-art ABA systems reason on the knowledge base directly instead of constructing the arguments (Lehtonen, Wallner, and Järvisalo 2021a; Lehtonen, Wallner, and Järvisalo 2021b). Indeed, many structured argumentation formalisms, including ABA, suffer from the drawback that the knowledge base gives rise to exponentially many (or even an infinite number of) arguments (Amgoud, Besnard, and Vesic 2014; Lehtonen, Wallner, and Järvisalo 2017; Strass, Wyner, and Diller 2019; Yun, Oren, and Croitoru 2020). On the other hand, this implies that the instantiated AF makes information within the knowledge base *explicit*, i.e., certain reasoning steps are performed during the AF construction.

The computational cost of the instantiation procedure can be divided into two steps: (1) the AF construction and (2) reasoning on the resulting AF, as illustrated in Figure 1. Since virtually all argumentative reasoning on ABA is NP-hard (Dimopoulos, Nebel, and Toni 2002), it is immediate that there is no hope for having both steps tractable in general. However, it turns out that in the standard instantiation procedure for ABA, *both* steps are intractable: the AF can be exponentially-sized and reasoning on the resulting AF can be NP-hard, thereby *paying the computational*

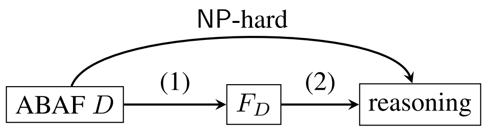

Figure 1: Sources for complexity in the instantiation procedure: (1) construction of the AF F_D; (2) reasoning in the AF F_D.

cost twice for the overall NP-hard reasoning tasks. This holds for instance for the prominent NP-complete problem of credulous acceptance under admissibility. Such an approach is a barrier, e.g., for utilizing AF solvers even if they are successful for AFs (Thimm and Villata 2017; Gaggl et al. 2020), since they have to operate on large AFs.

In this paper, we delve into the issue of the computational cost of the instantiation process and discuss a complexity trade-off: we show that we can confine the complexity to either step (1) or to step (2), with the other step being tractable. Based on two ABA fragments which we call atomic and symmetric ABA, we identify two approaches to achieve this:

1. by translating a general ABAF into an atomic ABAF, the size of the resulting AF is bounded, and
2. by translating a general ABAF into a symmetric ABAF, reasoning in the resulting AF is tractable.

We also empirically evaluate the first approach in practice, namely translating a general ABAF into an atomic ABAF, instantiating an AF from the atomic ABAF, and using the state-of-the-art AF solver MU-TOKSIA (Niskanen and Järvisalo 2020) to reason on the AF. We found that our novel approach is competitive, on instances with limited cyclic dependencies in rules, with the currently best performing ABA system, which directly works on the ABAF without instantiation (Lehtonen, Wallner, and Järvisalo 2021a). Our approach outperforms previous approaches to ABA reasoning that are based on instantiating an AF. In addition to foundational insights into the instantiation procedure, our approach thus also paves the way for modern AF solvers to compete with solvers that work without an instantiation procedure.

The main contributions of the paper is as follows.

- We first take care of instantiations into infinite AFs, and operate on the technical vehicle of *cores* of ABAFs, which serve as foundation for showing our results.
- We show that every ABAF can be translated to an atomic one so that the resulting instantiation leads to a polynomially-sized AF (w.r.t. the original ABAF).
- We present an approach that contains the complexity solely in the AF instantiation, by translating a general ABAF into a symmetric ABAF.
- We implement an approach that translates general ABAFs into atomic ABAFs and utilizes the state-of-the-art AF solver MU-TOKSIA (Niskanen and Järvisalo 2020). We compare our approach with the state of the art.

A preliminary version of this work was presented at the Non-Monotonic Reasoning workshop in 2022 (Rapberger, Ulbricht, and Wallner 2022). For proofs not contained in the paper and for the implementation we refer the reader to the supplementary material[1].

2 Background

We recall preliminaries for assumption-based argumentation (ABA) (Čyras et al. 2018; Bondarenko et al. 1997) and abstract argumentation frameworks (AFs) (Dung 1995).

Assumption-based Argumentation The first ingredient of ABA is a deductive system $(\mathcal{L}, \mathcal{R})$, with $\mathcal{L}$ a formal language and $\mathcal{R}$ a set of inference rules over $\mathcal{L}$. In this work we assume that $\mathcal{L}$ is a set of *atoms*. A rule $r \in \mathcal{R}$ is of the form $a_0 \leftarrow a_1, \dots, a_n$ with $a_i \in \mathcal{L}$. For such a rule we let $head(r) = a_0$, and $body(r) = \{a_1, \dots, a_n\}$. An ABA framework (ABAF) contains a deductive system and specifies which atoms are assumptions and what are their contraries.

Definition 1. *An ABAF is a tuple $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$, where $(\mathcal{L}, \mathcal{R})$ is a deductive system, $\mathcal{A} \subseteq \mathcal{L}$ a non-empty set of assumptions, and $\overline{}$ a function mapping assumptions $a \in \mathcal{A}$ to atoms $s \in \mathcal{L}$ (the contrary function).*

We extend the contrary function to sets: $\overline{S} = \{\overline{x} \mid x \in S\}$. In this work, we focus on ABAFs which are *flat*, i.e., no assumption can be derived: we require for each rule $r \in \mathcal{R}$ that $head(r) \notin \mathcal{A}$ holds. Moreover, we assume that ABA frameworks are finite ($\mathcal{L}$, $\mathcal{R}$, and $body(r)$ for each $r \in R$ are finite), and that each rule is stated explicitly, i.e., each rule is ground and does not contain any variables.

Arguments in an ABA framework $(\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ are based on proof trees. We refer to arguments based directly on proof trees as "tree-based arguments", because we will in later sections consider a different representation of arguments due to our computational purposes. In brief, a tree-based argument represents a derivation using rules in $\mathcal{R}$ to derive a claim s starting from a set of assumptions $S \subseteq \mathcal{A}$.

Definition 2. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF. A* tree-based argument *t based on D is a finite labeled rooted tree where i) the root is labeled with some atom $s \in \mathcal{L}$, ii) each leaf is labeled by an assumption $a \in \mathcal{A}$ or a dedicated symbol $\top \notin \mathcal{L}$, and iii) each internal node is labeled with $head(r)$ of a rule $r \in \mathcal{R}$ s.t. the set of labels of children of this node is equal to $body(r)$ or $\top$ if the body is empty.*

Note that if $a \in \mathcal{A}$, then the tree t consisting of a single node labeled with a is also a tree-based argument.

For a tree-based argument t we write $leaves(t)$ to denote the set of assumptions labeling the leaves; $cl(t)$ to denote the claim (or conclusion) of t, i.e., the label of the root; and $rules(t)$ to denote the set of rules required to construct t. We remark that there can be multiple tree-based arguments with the same set of assumptions and rules, and the same claim.

Remark 3. *Tree-based arguments are commonly denoted as derivations $S \vdash_R p$ where $S = leaves(t)$, $R = rules(t)$, and $p = cl(t)$. We will encounter a similar argument representation later on (we call them* core arguments*, cf. Section 4). To clearly distinguish both notions, we denote tree-based arguments with Latin letters (e.g.,t, u, or v).*

[1]https://bitbucket.org/lehtonen/acbar

For $S \subseteq \mathcal{A}$, we define derivability in ABA via

$$Th_D(S) = \{cl(t) \mid t \text{ tree-based argument}, leaves(t) \subseteq S\}.$$

That is, $Th_D(S)$ contains all atoms that can be derived (via tree-based arguments) using assumptions in S. We omit the subscript D if clear from the context.

We now recall conflicts, admissible sets, and subsequently the remaining semantics. A set A of assumptions attacks a set B of assumptions if it is possible to derive the contrary of some assumption in B from A.

Definition 4. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF and $A, B \subseteq \mathcal{A}$ be two sets of assumptions. Assumption set A attacks assumption set B in D if $\overline{b} \in Th(A)$ for some $b \in B$.*

This yields notions of conflict-freeness and defense.

Definition 5. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF. An assumption set $A \subseteq \mathcal{A}$ is conflict-free in D, $A \in cf(D)$, iff A does not attack itself; A defends assumption set $B \subseteq \mathcal{A}$ in D iff for all $C \subseteq \mathcal{A}$ that attack B it holds that A attacks C.*

Now we are ready to define the standard semantics.

Definition 6. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF and let $A \in cf(D)$. Then the set A is*

- admissible*, $A \in adm(D)$, iff A defends itself;*
- complete*, $A \in com(D)$, iff A is admissible and contains every assumption set defended by A;*
- preferred*, $A \in prf(D)$, iff A is admissible and there is no admissible set of assumptions B with $A \subset B$;*
- stable*, $A \in stb(D)$, iff A attacks each $\{x\} \subseteq \mathcal{A} \setminus A$.*

An atom s is credulously accepted under σ in an ABAF D iff there is a σ-assumption-set E s.t. $s \in Th_D(E)$.

Example 7. *Consider the ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ where*

$$\mathcal{L} = \{c_1, c_2, \phi\} \cup \mathcal{A},$$

$$\mathcal{A} = \{x_1, x_1', x_2, x_2'\} \text{ with } \overline{x_i} = x_i', \overline{x_i'} = x_i,$$

moreover, the rules $\mathcal{R}$ of the given ABA are

$$c_1 \leftarrow x_1; \quad c_1 \leftarrow x_2'; \quad c_2 \leftarrow x_1';$$

$$c_2 \leftarrow x_2; \quad \phi \leftarrow c_1, c_2.$$

It holds that each $A \subseteq \mathcal{A}$ is admissible whenever $\{x_i, x_i'\} \not\subseteq A$ for $i \in \{1, 2\}$ (no "complementary literals"). Moreover, the literal ϕ is credulously accepted under admissibility, since, e.g., $\{x_1, x_2\}$ is admissible and $\phi \in Th(\{x_1, x_2\})$.

Semantics of ABA frameworks can be alternatively defined in terms of arguments and attacks.

Abstract Argumentation An abstract argumentation framework (AF) (Dung 1995) is a directed graph $F = (\mathbb{A}, \mathbb{R})$ where $\mathbb{A}$ represents a set of arguments and $\mathbb{R} \subseteq \mathbb{A} \times \mathbb{A}$ models *attacks* between them. For two arguments x, y s.t. $(x, y) \in \mathbb{R}$ we say x *attacks* y. A set $E \subseteq \mathbb{A}$ attacks an argument x if there is some $y \in E$ that attacks x. Set E is *conflict-free* in F iff for no $x, y \in E$, $(x, y) \in R$; E *defends* an argument x if E attacks each attacker of x. A conflict-free set E is *admissible* in F ($E \in adm(F)$) iff it defends all its elements. A *semantics* is a function $F \mapsto \sigma(F) \subseteq 2^A$; each $E \in \sigma(F)$ is called a σ-*extension*. Here we consider so-called *complete*, *preferred*, and *stable* semantics (abbr. com, prf, and stb).

Definition 8. *Let $F = (\mathbb{A}, \mathbb{R})$ be an AF and $E \in adm(F)$. Then $E \in com(F)$ iff E contains all arguments it defends; $E \in prf(F)$ iff E is $\subseteq$-maximal in $com(F)$; $E \in stb(F)$ iff E attacks each argument $x \in \mathbb{A} \setminus E$.*

For a given AF $(\mathbb{A}, \mathbb{R})$ and $x \in \mathbb{A}$, it holds that x is credulously accepted in F w.r.t. a semantics σ iff there is a σ-extension E containing x.

Instantiation ABA semantics can be alternatively defined by constructing a suitable AF.

Definition 9. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF and let t and v be two tree-based arguments based on D. We say that t* attacks *v iff $cl(t) \in \overline{leaves(v)}$.*

Collecting all tree-based arguments and attacks based on D results in the AF F_D corresponding to the given ABAF.

Definition 10. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF. The pair $F_D = (\mathbb{A}, \mathbb{R})$ is called the AF corresponding to D if $\mathbb{A}$ is the set of all tree-based arguments based on D, and $\mathbb{R}$ is the set of all attacks based on D.*

Claims of tree-based arguments are collected via $cl(E) = \{cl(t) \mid t \in E\}$ for a set E of tree-based arguments, and assumptions via $asms(E) = \bigcup_{t \in E} leaves(t)$. We recall the relation between ABAFs and AFs (see, e.g., Čyras et al., 2018, Theorem 4.3).

Proposition 11. *Let D be an ABAF, F_D the associated AF, and $\sigma \in \{adm, com, prf, stb\}$. If $A \in \sigma(D)$, then*

$$\{t \mid leaves(t) \subseteq A, t \text{ is a tree-based argument}\} \in \sigma(F_D).$$

If $E \in \sigma(F_D)$, then $asms(E) \in \sigma(D)$.

To clearly distinguish semantics, we say that $A \subseteq \mathcal{A}$ is an assumption set under semantics σ (or a σ-assumption-extension) and that a set of tree-based arguments E is an extension under semantics σ (or σ-extension for short).

We remark that credulous acceptance of tree-based arguments in a given AF can be directly generalized to ask for acceptance of claims s of tree-based arguments, i.e., asking whether there is some σ-extension containing some tree-based argument t with claim s.

Complexity results for reasoning in ABA and AFs were established (see, e.g., the chapter by Dvořák and Dunne (2018) for an overview) when the corresponding structure is given (in particular for AFs the full AF is given as input). For both assumption sets and extensions, deciding credulous acceptance under admissible, preferred, complete, and stable semantics is NP-complete.

3 Complexity Trade-Offs

Let us delve into the computational machinery underlying the instantiation procedure. We identify two sources of complexity when reasoning in ABA via instantiating an AF.

1. Construction of the AF corresponding to the ABAF.
2. The computational complexity of reasoning in the AF.

We illustrate this in Figure 1. Since reasoning in ABA is NP-hard, it is clear that both steps in (1) and (2) can not be performed together in polynomial time. However, according to the original definition of ABA, *neither* of these steps is tractable. Let us now inspect both points in detail.

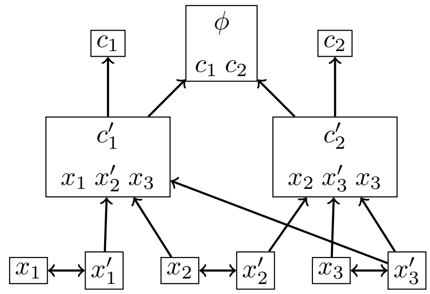

Figure 2: AF instantiation of the ABA framework from Reduction 14 for the formula $\phi = (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_3)$.

Size of the constructed AF. Let us start with the first step. It is folklore that the corresponding AF to an ABAF is infinite in general. We say that an AF $F = (\mathbb{A}, \mathbb{R})$ is infinite if $\mathbb{A}$ is infinite. An AF F is finitary (Dung 1995) if it holds that each argument $t \in \mathbb{A}$ is attacked by a finite number of arguments (but the overall number of arguments may still be infinite). It is not hard to construct an ABAF s.t. the induced AF F_D becomes infinite.

Example 12. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework with $\mathcal{A} = \{a, b\}$, $\mathcal{L} = \{x, y\} \cup \mathcal{A}$, four rules $(x \leftarrow a)$, $(x \leftarrow x)$, $(y \leftarrow y)$, and $(y \leftarrow b)$, and $\overline{a} = y$ and $\overline{b} = x$. There are infinitely many tree-based arguments based on D (via chaining rules arbitrary many times), and all arguments with claim x are attacked by all tree-based arguments concluding y (of which there are infinitely many).*

This leads to the simple observation stated next.

Observation 13. *Given an ABA framework, the corresponding AF can be infinite and non-finitary.*

This observation implies that step (1) in Figure 1 is in general not just exponential, but even infinite.

Reasoning in the Constructed AF. Now suppose we are given F_D. Is it now the case that reasoning in F_D is easy, because all the computational effort was done due to constructing F_D? Unfortunately, the answer is negative. Even with F_D given, it is still NP-hard (in the size of F_D) to decide the standard reasoning tasks.

To demonstrate this, let us consider the following (direct) reduction from SAT.

Reduction 14. *Let ϕ be a Boolean formula in CNF over clauses C and variables X. Let $X' = \{x' \mid x \in X\}$ and $C' = \{c' \mid c \in C\}$. We construct $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ with*

- *$\mathcal{L} = X \cup X' \cup C \cup C' \cup \{\phi\}$,*
- *$\mathcal{A} = X \cup X' \cup C$,*
- *$\overline{x} = x'$ and $\overline{x'} = x$ for each $x \in X$, and $\overline{c} = c'$ for each $c \in C$, and*
- *let the set of rules be composed of $\phi \leftarrow C$, as well as $c' \leftarrow \{l' \mid l \in c\} \cup \{l \mid \neg l \in c\}$ for each $c \in C$.*

An example of this reduction is depicted in Figure 2.

The corresponding ABAF F_D can be constructed in polynomial time; indeed, it can be checked that the AF F_D contains $2 \cdot (|X| + |C|) + 1$ many tree-based arguments. Moreover, the resulting AF corresponds to the *standard translation* for AFs (Dvořák and Dunne 2018, Reduction 3.6) (including further auxiliary arguments corresponding to the assumptions $c \in C$). As discussed by Dvořák and Dunne (2018), the formula ϕ is satisfiable iff the argument t with claim ϕ is acceptable w.r.t. each of the semantics considered in this paper. We arrive at the following observation.

Observation 15. *Credulous reasoning under semantics $\sigma \in \{adm, com, prf, stb\}$ in AFs is* NP*-complete in the class of (finite) AFs stemming from instantiating an ABAF.*

We therefore conclude that step (2) is also intractable.

Outline. We set out to consider all aspects raised in this section: (i) unbounded number of arguments (Section 4), (ii) intractability in the AF construction step (Section 5), and (iii) intractability in the reasoning step (Section 6).

4 Argument Representation

In this section, we discuss argument representations in ABA. First, we will take a closer look at tree-based arguments. Even though their number is not bounded in general, we establish a bound on the number of tree-based arguments depending on the derivation-depth and the length of the rules. Next, we show that it suffices to consider the assumptions and the claim of ABA arguments. That is, we abstract away from the particular set of rules and consider arguments of the form (A, s) where A is the set of assumptions and s is the claim of a tree-based argument t. Most importantly, we will show that both representations are semantically equivalent.

4.1 Tree-based Arguments

As already observed in Section 3, a finite ABAF can give rise to infinitely many tree-based arguments. We show that, for the finite case, the number of tree-based arguments depends on certain parts of the input. More precisely, we establish a bound on the number of tree-based arguments that considers derivation-depth and rule-size of a given ABAF.

Formally, an ABAF D is bounded by k-derivation-depth if for each proof tree t of D it holds that the tree has height at most k (i.e., the longest path from an assumption to the claim is at most k). A rule $h \leftarrow b_1, \ldots, b_n$ is bounded by m if $n \leq m$, and an ABA D is rule-size bounded by m if each rule in D is bounded by m.

The number of tree-based arguments that can be constructed from a given ABAF D depends on the number of rules, derivation-depth, rule-size, and number of rules with the same heads, as follows.

Proposition 16. *For each m-rule-size bounded ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ with $|\{r \in \mathcal{R} \mid head(r) = s\}| \leq l$ for all $s \in \mathcal{L}$, there are at most*

$$l^p \cdot |\mathcal{L} \setminus \mathcal{A}|, \quad \textit{with } p = \sum_{i=0}^{k-1} m^i,$$

many tree-based arguments of height $k \geq 1$.

Thus, the number of tree-based arguments in an ABAF that is bounded by k-derivation-depth is $|\mathcal{A}|$ plus the sum of all tree-based arguments of height $j \leq k$. Hence, exponentiality of the number of tree-based arguments stems from p, i.e., the derivation-depth and rule-size.

If the derivation-depth is equal to one, i.e., if $k = 1$, then the number of tree-based arguments depends linearly on D and is independent of the length of the rules. In this case, it holds that $p = 1$, hence there are at most $l \cdot |\mathcal{L} \setminus \mathcal{A}| + |\mathcal{A}|$ many tree-based arguments.

Bounding the rule-size does not yield similar advantages: for $m = 1$ the number of arguments still depends on the derivation-depth and there might be exponentially many, more precisely, up to $l^k \cdot |\mathcal{L} \setminus \mathcal{A}|$ many tree-based arguments of height k. We observe that bounding either the derivation-depth by some $k > 1$ or the rule-size individually does not suffice to prevent a potential exponential blowup.

Moreover, while it is guaranteed that, given a fixed ABAF D, the rule size is bounded by some number m (depending on D) by definition, the derivation-depth of an ABAF can be infinite (as is the case in Example 12). Hence, as outlined in the previous section, an ABAF can have potentially infinitely many tree-based arguments.

4.2 Core Arguments

As one can see in Example 12, AFs corresponding to an ABAF can be "cut down" to a finite core by removing "duplicates" of arguments. This observation is sometimes assumed in the literature and stated for other forms of structured argumentation (Amgoud, Besnard, and Vesic 2014).

Tree-based arguments in an ABAF are defined as proof trees, with each derivation t based on a set of assumptions $A = \mathit{leaves}(t)$ and a claim $s = \mathit{cl}(t)$. While the rules in $\mathit{rules}(t)$ are driving derivability, they are not important when evaluating arguments: conflicts between arguments are solely specified via the assumptions A and claim s. A natural way to represent arguments is thus by using only A and s. Motivated by this observation, we identify arguments with pairs (A, s) but insist that there is a corresponding proof tree t with $\mathit{leaves}(t) = A$ and $\mathit{cl}(t) = s$ in the given ABAF. We call the resulting set of tuples the *core* of an ABAF. Let $A, B \subseteq \mathcal{A}$ and $s, t \in \mathcal{L}$. We say that (A, s) attacks (B, t) in an ABAF D if $\exists b \in B$ s.t. $\overline{b} = s$.

Definition 17. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ \ })$ be an ABAF. Define $\mathbb{A} = \{(\mathit{leaves}(t), \mathit{cl}(t)) \mid t$ is a tree-based argument in $D\}$ and $\mathbb{R}$ as the set of all attacks between arguments in $\mathbb{A}$. The AF $F = (\mathbb{A}, \mathbb{R})$ is called the core of D.*

Claims and assumptions of a set of arguments $H \subseteq \mathbb{A}$ are defined similarly as for tree-based arguments: $\mathit{cl}(H) = \{s \mid (A, s) \in H\}$ and $\mathit{asms}(H) = \bigcup_{(A,s) \in H} A$. Given an ABAF D, let F be the corresponding AF and F' the core.

- for each tree-based argument t in F there is an argument $(\mathit{leaves}(t), \mathit{cl}(t))$ in F' and vice versa, and thus
- for each $\{\mathit{adm}, \mathit{com}, \mathit{prf}, \mathit{grd}, \mathit{stb}\}$, there is some $H \in \sigma(F)$ with $\mathit{cl}(H) = S$ and $\mathit{asms}(H) = A$ iff there is some $H' \in \sigma(F')$ with $\mathit{cl}(H') = S$ and $\mathit{asms}(H') = A$.

If one is not interested in the actual derivation of a claim, representing an argument as a pair (A, s) narrows down the argument to the information required in order to reason in the corresponding AF (the core). A simple observation is that the core is always finite. Thus, this modification ensures that step (1) in Figure 1 is now finite, although not tractable.

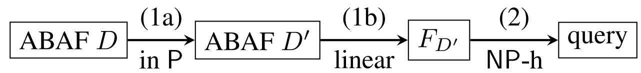

Figure 3: How to shift intractability to reasoning in the AF: (1a) construct semantics-preserving ABAF D' (in P) s.t. (1b) D' can be instantiated in linear time; reasoning in $F_{D'}$ remains NP-hard.

Observation 18. *The core F_D of an ABAF D has at most $|2^{\mathcal{A}}| \cdot |\mathcal{L} \setminus \mathcal{A}| + |\mathcal{A}|$ arguments.*

To some extent surprising perhaps, we can find a complexity-theoretic result supporting the intuition that the core is a more compact representation: while deciding whether a proof tree constitutes a tree-based argument for a given ABA is immediate, it is NP-hard to decide whether a pair (A, s) occurs in the core.

Proposition 19. *It is NP-hard to decide whether there is a proof tree from a given set of assumptions to a given claim.*

5 Transforming ABAFs: Size of F_D

The goal of this section is to transform the given ABAF D in a way that the construction of the AF, i.e., step (1) in Figure 1, is tractable and thus, the main source of complexity is the reasoning within the AF, i.e., lies in step (2).

To this end we first require a sub-class of ABAFs facilitating our endeavor. This leads to the notion of so-called *atomic* ABAFs which require that each rule body consists of assumptions only. In such an atomic ABAF, constructing the AF, i.e., step (1), can be achieved in polynomial time. The crucial result in this section is Theorem 27 stating that we can translate any given ABAF into an atomic one, without increasing the size exponentially. Figure 3 illustrates the overall procedure: for a given ABAF D, we construct a corresponding ABAF D' in polynomial time (step (1a)) and construct the corresponding AF $F_{D'}$ in linear time (step (1b)). Hence, the source of complexity lies solely in step (2).

The main result of this section is then that for each general ABAF D we can

1. find an atomic ABAF D' that preserves the considered semantics under projection and
2. has polynomially many, concretely $|\mathcal{A}| + |\mathcal{L} \setminus \mathcal{A}| \cdot (|\mathcal{R}| + 3)$ many arguments.

We proceed in two steps: after defining atomic ABAFs, we first show how to treat ABAFs satisfying a certain acyclicity constraint and translate them into atomic ABAFs. After that we extend the approach to circular ABAFs.

5.1 Atomic ABAFs

An atomic ABAF is composed of rules which have only assumptions in their bodies.

Definition 20. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ \ })$ be an ABAF. A rule $r \in \mathcal{R}$ is called* atomic *if $\mathit{body}(r) \subseteq \mathcal{A}$. The ABAF D is called* atomic *if each rule $r \in \mathcal{R}$ is atomic.*

An example atomic ABAF is shown in Figure 2.

Recall that tree-based arguments are constructed inductively, by utilizing given arguments $A_1, \dots, A_n$ having conclusions $p_1, \dots, p_n$ in combination with a rule r s.t. $body(r) = \{p_1, \dots, p_n\}$. However, since our ABAFs are assumed to be flat, i.e., no assumption occurs in any rule head, argument construction in an atomic ABAF consists of the base case only: each rule induces exactly one argument and no argument can be constructed by combining different rules or arguments. Hence the derivation-depth of each atomic ABAF is one (cf. Section 4). The set of arguments corresponding to an atomic ABAF is given as follows:

- Assumptions a induce arguments $(\{a\}, a)$.
- Rules $head(r) \leftarrow body(r)$ induce $(body(r), head(r))$.

This yields the following observation about the class of atomic ABAFs.

Proposition 21. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an atomic ABAF. The core F_D of D has $|\mathcal{A}| + |\mathcal{R}|$ many arguments and can be computed in polynomial time.*

In this result, one can interchange cores and tree-based arguments. We want to emphasize though that reasoning in the core of an atomic ABAF remains NP-hard. That is, atomic ABAFs preserve the full complexity of reasoning.

Theorem 22. *Credulous reasoning under semantics $\sigma \in \{adm, com, prf, stb\}$ is NP-complete for atomic ABAFs.*

5.2 From Non-Circular ABA to Atomic ABA

We show how to translate a given ABAF into an atomic one. For ease of presentation we first assume that our given ABAF is not *circular*; we show later how to generalize our approach. We take the definition of circular tree-based arguments (proof trees) from Craven and Toni (2016).

Definition 23. *A tree-based argument is circular if there is a path from a leaf to the root which contains two distinct vertices with the same label. An ABAF is circular if there is a circular tree-based argument for this framework.*

Now we are ready to translate a given non-circular ABAF into an atomic one. Our transformation is defined as follows. Intuitively, for each conclusion s derivable from a given ABAF D, we introduce fresh assumptions s_d (s is derivable) and s_{nd} (s is not derivable) that simulate tree-derivations. The resulting ABAF D' is atomic.

Definition 24. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be a non-circular ABA framework such that each $s \in \mathcal{L}$ is in $Th_D(\mathcal{A})$. We define $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}', \overline{}')$ as the AF-sensitive ABA framework of D as follows. For each $s \in \mathcal{L} \setminus \mathcal{A}$*

- *let s_d and s_{nd} be two fresh assumptions, with*
- *$\overline{s_d} = s_{nd}$ and $\overline{s_{nd}} = s$ in $\overline{}'$.*

Let $\mathcal{A}' = \mathcal{A} \cup \{s_d, s_{nd} \mid s \in \mathcal{L} \setminus \mathcal{A}\}$ and $\mathcal{L}' = \mathcal{L} \cup \mathcal{A}'$. The assumptions that are present in D have the same contraries in D' as in D. For each rule $r \in \mathcal{R}$, let r' be r except that if $body(r)$ contains a non-assumption s, replace s by s_d. Finally, set $\mathcal{R}' = \{r' \mid r \in \mathcal{R}\}$.

The fact that D' is atomic is immediate by definition.

Proposition 25. *Given a non-circular ABAF D, the AF-sensitive ABAF D' is atomic.*

The following example illustrates our construction.

Example 26. *Consider an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ with assumptions $\mathcal{A} = \{a, b\}$, rules $\mathcal{R}$:*

$$r_1 : p \leftarrow q; \quad r_2 : q \leftarrow a; \quad r_3 : r \leftarrow b;$$

and contraries $\overline{a} = r$ and $\overline{b} = p$. In D, both $\{a\}$ and $\{b\}$ are admissible as they symmetrically attack each other.

Following Definition 24, we obtain the corresponding ABA $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}', \overline{}')$ with assumptions a, b, and additional assumptions p_d, p_{nd}, q_d, q_{nd}, r_d, r_{nd}; and rules $\mathcal{R}$:

$$r'_1 : p \leftarrow q_d; \quad r'_2 : q \leftarrow a; \quad r'_3 : r \leftarrow b.$$

The assumption q_d is defended by each assumption set that derives q (since q_d is attacked by q_{nd} which is in turn attacked by all assumption sets that derive q). Consequently, $\{q_d, a\}$ is admissible since it derives q and p and thus defeats the attackers b and q_{nd}. Likewise, $\{b, q_{nd}\}$ is admissible in D' as it defends itself against the attack from q_d and a.

As we have seen in the above example, restricting the outcome to the initial set of assumptions yields the original extensions. This is not a coincidence: acceptable assumption sets and derivations are preserved when projecting to $\mathcal{A}$ of the original ABA framework.

Theorem 27. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be a non-circular ABA framework such that each $s \in \mathcal{L}$ is in $Th_D(\mathcal{A})$ and D' the AF-sensitive ABA framework of D. It holds that*

- *if $E \in adm(D)$, then there is an $E' \in adm(D')$ with $E = E' \cap \mathcal{A}$ and $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E') \cap \mathcal{L}$, and*
- *if $E' \in adm(D')$, then $E' \cap \mathcal{A} \in adm(D)$ and $Th_{\mathcal{R}}(E) \supseteq Th_{\mathcal{R}'}(E') \cap \mathcal{L}$.*

Moreover, for $\sigma \in \{com, prf, stb\}$ we find that

- *if $E \in \sigma(D)$, then there is an $E' \in \sigma(D')$ with $E = E' \cap \mathcal{A}$ and $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E') \cap \mathcal{L}$, and*
- *if $E' \in \sigma(D')$, then $E' \cap \mathcal{A} \in \sigma(D)$ and $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E') \cap \mathcal{L}$.*

5.3 Transforming Circular ABAFs

As indicated, we do not have to restrict to non-circular ABAFs. In this subsection we show that any ABAF can be translated into a non-circular one (of polynomial size).

The underlying idea is to disrupt cycles using auxiliary rules and atoms. To this end, we utilize the following crucial observations. For a given ABAF D,

1. it suffices to construct tree-based arguments up to a certain derivation-depth k to preserve the semantics; and
2. k depends polynomially on D, i.e., there is a polynomial p such that $k < p(|D|)$ for any ABAF D.

After establishing both observations, we define an ABAF D' where we simulate the k derivation steps with auxiliary rules. Intuitively, the procedure is as follows: we copy each rule k times and introduce a hierarchical order between atoms on different levels. In this way, we guarantee that an atom s_i which appears in some rule r_i (on the i-th level) can participate in deriving only atoms s_j with $j > i$, thus preventing cyclic derivations.

Bounding the derivation-depth Towards our results, we make use of the concept of *redundant arguments*.

Definition 28. *Let D be an ABA framework and $F = (\mathbb{A}, \mathbb{R})$ the core of D. An argument $(A, s) \in \mathbb{A}$ is called* redundant *in F iff there is an argument $(A', s) \in \mathbb{A}$ with $A' \subsetneq A$.*

The following lemma formalizes that we can remove redundant arguments from the core without changing the semantics of the induced AF F. As usual for AFs, by $F{\downarrow_S}$ we define the projection $F{\downarrow_S} = (\mathbb{A} \cap S, \mathbb{R} \cap (S \times S))$.

Lemma 29. *Let D be an ABAF and $F = (\mathbb{A}, \mathbb{R})$ the core of D. Let $x = (A, s) \in \mathbb{A}$ be a redundant argument. Then for each $\sigma \in \{adm, com, prf, stb\}$ we have*

$$\{cl(E) \mid E \in \sigma(F)\} = \{cl(E) \mid E \in \sigma(F{\downarrow_{\mathbb{A}\setminus\{x\}}})\}.$$

If we are only interested in arguments that are non-redundant, then we can skip the computation of many tree-based arguments, because new information can only be found up to a certain derivation depth. Each non-redundant argument can be obtained with derivation depth $|\mathcal{L} \setminus \mathcal{A}|$.

Proposition 30. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework and $F = (\mathbb{A}, \mathbb{R})$ the core of D. If $(A, s) \in \mathbb{A}$ is not redundant, then there is some tree-based argument t with $leaves(t) = A$ and $cl(t) = s$ in D such that the derivation depth of t is at most $|\mathcal{L} \setminus \mathcal{A}|$.*

We have shown that (1) it is possible to bound the derivation-depth by some k without changing the semantics of a given ABAF; and (2) $k = |\mathcal{L} \setminus \mathcal{A}|$ is linear.

Disrupting cycles Let us continue with our procedure for constructing a non-circular ABAF from a circular one.

Definition 31. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF. We define the non-circular ABA D° as follows. Let $k = |\mathcal{L} \setminus \mathcal{A}|$. For each atomic rule $r = s \leftarrow p_1, \dots, p_n$ in $\mathcal{R}$ we consider k copies $r_1, \dots, r_k$ of the form $r_i = s^i \leftarrow p_1, \dots, p_n$ with $s^k = s$. For each remaining rule $r = s \leftarrow p_1, \dots, p_n$ in $\mathcal{R}$ we consider $k - 1$ copies $r_2, \dots, r_k$ where for each $2 \leq j \leq k$: $head(r_j) = s^j$ with $s^k = s$, if $p_i \in \mathcal{A}$, then $p_i \in body(r_j)$, if $p_i \notin \mathcal{A}$, then $p_i^{j-1} \in body(r_j)$. We let $D^\circ = (\mathcal{L}^\circ, \mathcal{R}^\circ, \mathcal{A}, \overline{})$ where*

$$\mathcal{L}^\circ = \mathcal{L} \cup \bigcup_{j=1}^{k} \{s^j \mid s \in \mathcal{L} \setminus \mathcal{A}\} \text{ and}$$

$$\mathcal{R}^\circ = \bigcup_{j=1}^{k} \{r^j \mid r \text{ atomic }\} \cup \bigcup_{j=2}^{k} \{r^j \mid r \text{ not atomic }\}.$$

Example 32. *Consider an ABAF with $\mathcal{A} = \{a\}$, $\mathcal{L} = \{a, p, q\}$, and rules $\mathcal{R} = \{p \leftarrow a,\ p \leftarrow q,\ q \leftarrow p\}$. We have $k = 2$ and thus obtain 4 rules $p^1 \leftarrow a$, $p^2 \leftarrow a$, $p^2 \leftarrow q^1$, and $q^2 \leftarrow p^1$, where $p^2 = p$ and $q^2 = q$.*

The following lemma shows that D° captures all non-redundant arguments in D.

Lemma 33. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF. If there is a non-redundant argument t in D then there is also an argument t' in D° with $leaves(t') = leaves(t)$ and $cl(t') = cl(t)$. Vice versa, if t is an argument in D°, then there is also an argument t' in D satisfying the same conditions.*

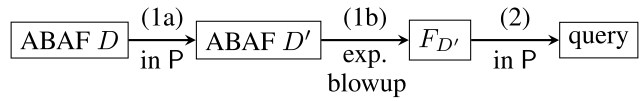

Figure 4: Shifting intractability to reasoning in the AF: (1a) construct acceptance-preserving ABAF D' (in P) s.t. (1b) D' has a simple structure; reasoning in $F_{D'}$ is in P.

Since D° is non-circular by construction, we obtain the following desired corollary.

Corollary 34. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA. Then D° is non-circular and satisfies $\sigma(D) = \sigma(D^\circ)$ for each $\sigma \in \{adm, com, prf, grd, stb\}$.*

5.4 From ABA to Atomic ABA

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

Theorem 35. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF. There is an ABAF D' s.t. $F_{D'}$ has at most $|\mathcal{A}| + |\mathcal{L} \setminus \mathcal{A}| \cdot (|\mathcal{R}| + 3)$ many arguments and D' preserves the σ-extensions of D under projection.*

Proof. First, we apply the construction from Definition 31. We construct $D^\circ = (\mathcal{L}^\circ, \mathcal{R}^\circ, \mathcal{A}, \overline{})$ in time $\mathcal{O}(|D|^2)$ by looping through all rules, copying them $|\mathcal{L} \setminus \mathcal{A}|$ times, and adding rules $s \leftarrow s^j$ for all newly introduced atoms s^j. The ABAF D° contains $|\mathcal{L} \setminus \mathcal{A}|(|\mathcal{R}| + 1)$ many rules. By Corollary 34, the transformation preserves the semantics. Next, we apply the construction from Definition 24 and obtain an atomic ABAF $(D^\circ)' = ((\mathcal{L}^\circ)', (\mathcal{R}^\circ)', \mathcal{A}', \overline{}')$ in linear time in the size of D°. We obtain $|\mathcal{A}| + 2|\mathcal{L} \setminus \mathcal{A}|$ many assumptions. Note that we do not add further rules, i.e., $|(\mathcal{R}^\circ)'| = |\mathcal{R}^\circ|$. By Theorem 27, the semantics are preserved under projection. Moreover, by Proposition 21, the number of arguments in $(D^\circ)'$ is equal to $|\mathcal{A}'| + |(\mathcal{R}^\circ)'| = |\mathcal{A}'| + |\mathcal{R}^\circ| = |\mathcal{A}| + 2|\mathcal{L} \setminus \mathcal{A}| + |\mathcal{L} \setminus \mathcal{A}|(|R| + 1)$. ☐

6 Transforming ABAFs: Reasoning in F_D

In this section we show how to transform an ABAF D in a way that step (2) in Figure 1 is tractable (w.r.t. the size of the constructed AF) and thus, the main source of complexity lies in the size of the AF, i.e., step (1). Figure 4 illustrates the idea: from a given ABAF D, we construct a corresponding ABAF D' in polynomial time (step (1a)) which yields an AF of exponential size but with simple structure (step (1b)).

Again, we require a suitable ABA sub-class to achieve our goal. To this end we introduce a novel class called *symmetric* ABAFs. They are tailored to ensure that reasoning is tractable (w.r.t. the size of F_D), but the number of arguments is exponential in general.

6.1 Symmetric ABAFs

An ABAF is symmetric if the contraries are symmetric.

Definition 36. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABAF. We call D* symmetric *whenever $\overline{a} = b$ iff $\overline{b} = a$ for all $a, b \in \mathcal{A}$.*

First we observe that credulous reasoning is still NP-hard, even if we restrict to symmetric ABAFs.

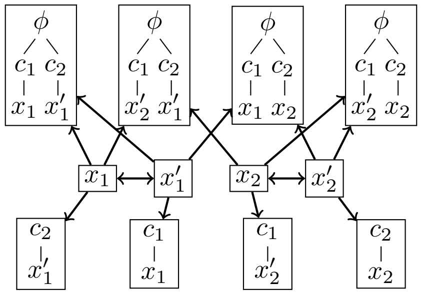

Figure 5: AF instantiation of the ABA framework from Example 38 for the formula $\phi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$ (cf. Reduction 37).

Reduction 37. *Let $\phi = c_1 \wedge \cdots \wedge c_m$ be a Boolean formula in conjunctive normal form (CNF) over clauses $C = \{c_1, \ldots, c_m\}$ and Boolean variables $X = \{x_1, \ldots, x_n\}$. Define $X' = \{x' \mid x \in X\}$. Construct $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ by $\mathcal{L} = X \cup X' \cup C \cup \{\phi\}$, $\mathcal{A} = X \cup X'$, $\overline{x} = x'$ and $\overline{x'} = x$ for each $x \in X$, and let the set of rules be composed of $\phi \leftarrow c_1, \ldots, c_m$, and $c_i \leftarrow z$ with $z = x$ and $x \in c_i$ or $z = x'$ and $\neg x \in c_i$.*

Example 38. *Given the CNF-formula $\phi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$. Following Reduction 37, we obtain an ABAF $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ which contains the assumptions $\mathcal{A} = \{x_1, x_2, x_1', x_2'\}$ and the rules $\varphi \leftarrow c_1, c_2$, $c_1 \leftarrow x_1$, $c_1 \leftarrow x_2'$ (since $x_1, \neg x_2 \in c_1$) as well as $c_2 \leftarrow x_1'$, $c_2 \leftarrow x_2$ (since $x_1', x_2 \in c_2$). Moreover, the ABA framework assigns symmetric contraries, i.e., $\overline{x_i} = x_i'$ and $\overline{x_i'} = x_i$ for $i \in \{1, 2\}$. The induced AF F_D is depicted in Figure 5.*

Complexity of reasoning in symmetric ABAFs follows.

Proposition 39. *Credulous reasoning under semantics $\sigma \in \{adm, com, prf, stb\}$ is* NP*-complete for symmetric ABAFs.*

Hence, similar to atomic ABAFs, symmetric ABAFs have the full computational hardness of general (flat) ABAFs.

On the other hand, we observe that the computational hardness in symmetric ABAs stems entirely from the construction of arguments: constructing the cores results in AFs with $|\mathcal{A}|/2$ even cycles of length 2 (a cycle for every assumption and its negation) satisfying that all arguments with claim $s \notin \mathcal{A}$ have only incoming attacks.

In Example 38, it is direct to check whether ϕ is the conclusion of an acceptable argument, once the AF F_D is given. Credulous reasoning in such AFs is decidable in polynomial time in the number of arguments, since it suffices to check if there exists an argument having the queried claim that is not attacked by both arguments in a 2-cycle.

Proposition 40. *Credulous reasoning under semantics $\sigma \in \{adm, com, prf, stb\}$ is decidable in polynomial time in cores of symmetric ABAs.*

Making use of symmetric ABAFs, we get the desired computational shift to step (1) in Figure 1.

6.2 From SAT to Symmetric ABAFs

We show a way of utilizing symmetric ABAFs to contain hardness in the AF construction. Suppose we want to transform an ABAF D. Since the SAT problem is NP-complete, we can construct in polynomial time a formula ϕ which is satisfiable iff p is credulously accepted in D. Now, Reduction 37 translates ϕ into a symmetric ABAF. All these steps can be performed in polynomial time.

Theorem 41. *Let $\sigma \in \{adm, com, prf, stb\}$. For each ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ and $p \in \mathcal{L}$ one can construct an ABA framework D' in polynomial time s.t. (1) x is credulously accepted in D' iff x is credulously accepted in D w.r.t. σ, and (2) credulous acceptance is in P under σ in the corresponding AF F.*

7 Experiments

We implemented the translation discussed in Section 5 to empirically evaluate the potential of answering credulous reasoning in ABA with a polynomially bounded AF construction. We implemented two procedures for breaking cycles in circular ABAFs. We call the first the *Naive* method: it is explained in Section 5.3, each rule is copied k times for a maximum derivation-depth k. We additionally implement a procedure, which we call the *SCC* method. The idea is similar, but the *SCC* method only copies rules within each strongly connected component (SCC) of the graph corresponding to the rules of a given ABAF, and up to the size of the particular SCC in terms of atoms. The implementation, written in Python, is available at https://bitbucket.org/lehtonen/acbar.

We ran the experiments on 2.60-GHz Intel Xeon E5-2670 57-GB machines with RHEL 8 under a per-instance time limit of 600 seconds and memory limit of 16 GB. We used the state-of-the-art AF solver MU-TOKSIA (Niskanen and Järvisalo 2020) for deciding credulous acceptance in the AF resulting from our translation. We compared our system against ASPFORABA, the state-of-the-art solver for ABA, which answers credulous acceptance with ASP encodings directly on the ABA level (Lehtonen, Wallner, and Järvisalo 2021a), as well as a state-of-the-art translation-based approach to ABA reasoning, ABA2AF (Lehtonen, Wallner, and Järvisalo 2017). We tested credulous acceptance under stable and admissible semantics (the latter coincides with credulous acceptance under complete and preferred semantics).

We considered non-circular and circular instances separately, creating two synthetic benchmark sets. To construct non-circular instances, we generated 10 ABA frameworks per each combination of the following parameters: number of atoms $N \in \{1000, 2000, 3000, 4000, 5000\}$, the proportion of the atoms that are assumptions $A_r \in \{0.15, 0.3, 0.7\}$, and the number of rules that have a given atom as a head and the rule width, i.e., number of atoms in the body of a given rule was selected uniformly at random from the interval $[1, n]$ for $n \in \{2, 5, 8, 13\}$. To enforce non-circularity, we selected a random permutation $(x_i)_{0<i\leq n}$ of the non-assumption atoms in the instance and let assumptions have index 0. When creating a rule deriving the atom x_i, we only allowed an atom x_j in the body if $j < i$. Any graph corresponding to the possible derivations in the ABAF created in this manner obeys the topological order specified by the chosen permutation and is thus a directed acyclic graph. Further,

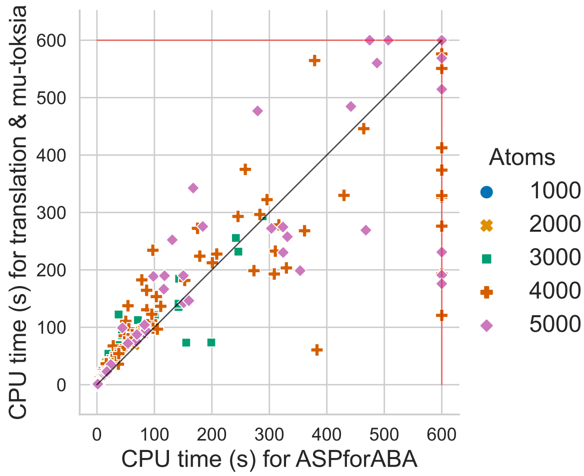

Figure 6: Per-instance runtime comparison of credulous reasoning between ASPFORABA and our novel ABA translation with the AF solver mu-toksia under *com* for non-circular instances.

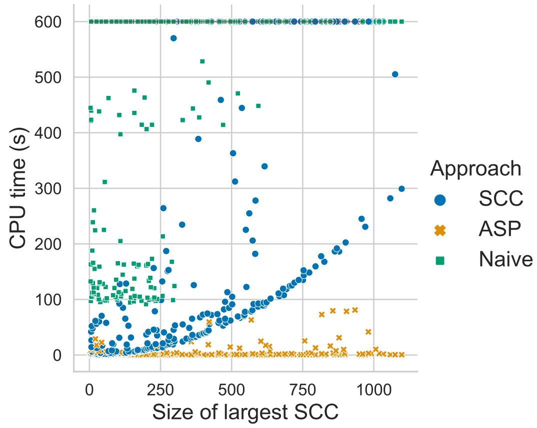

Figure 7: Per-instance runtimes of credulous acceptance in circular ABA frameworks in terms of the largest SCC in a given instance under stable semantics.

all assumptions were assigned a contrary at random. Finally, we randomly selected one non-premise atom per framework as a query, for a total of 2400 instances.

For the circular setting we fixed the ratio of assumptions to 0.3, and both the maximum number of rules per head and rule width to 5, to focus on the effect of the size of SCCs in the rules of a given instance. We generated 100 frameworks for each $N \in \{1000, 2000, 3000, 4000\}$ and selected a query at random, for a total of 400 instances. We let a proportion (selected uniformly at random from $[0, 1/50]$ for each framework) of the atoms later in the ordering than the head of a given rule be available for the body of the rule.

A comparison between ASPFORABA and our approach of translating an ABA framework to an atomic one before constructing an AF and deciding credulous acceptance via an AF solver is shown in Figure 6 for non-circular benchmarks under complete semantics, with timeouts shown as 600 seconds. We used the AF solver MU-TOKSIA for our approach. Our approach outperforms ASPFORABA in terms of timeouts, while for many apparently easier instances, ASPFORABA is faster. For credulous acceptance under stb, we observed similar results, but with our translation-based approach and ASPFORABA tied in number of timeouts.

In contrast, ABA2AF, an approach that employs a translation from ABA to AF and using an AF solver, performs significantly worse. ABA2AF is able to solve only 195 out of the 2400 acyclic instances, compared to 2371 for our translation-based approach. This confirms the efficiency gains from preprocessing an ABA framework to an atomic one, ensuring that the resulting AF is of polynomial size.

A comparison between our approach, using the different cycle disruption techniques, and ASPFORABA is shown in Figure 7. The instances are ordered according to the largest SCC occurring in a graph corresponding to the rules of the instance. When using the *Naive* cycle disruption method, our approach is outperformed on all instances. However, using the *SCC* cycle disruption method improves our approach considerably. The runtime of our approach using the *SCC* method has a clear correlation with the size of the largest SCC of a given instance. When the size of the SCCs in an instance is limited (especially to under around 200 atoms), our approach with the *SCC* method is competitive with ASPFORABA. As the size of the largest SCC in an instance grows, the runtime of our approach deteriorates in comparison to ASPFORABA.

8 Discussion

The computation of an AF from structured argumentation formalisms, and different argument representations and optimizations, have been considered before for ABA (Craven and Toni 2016; Bao, Čyras, and Toni 2017; Lehtonen, Wallner, and Järvisalo 2017), and for other forms of structured argumentation (Amgoud, Besnard, and Vesic 2014; Yun, Vesic, and Croitoru 2018). Moreover, complexity of ABA has been investigated in several directions (Dimopoulos, Nebel, and Toni 2002; Čyras, Heinrich, and Toni 2021; Lehtonen, Wallner, and Järvisalo 2021a; Karamlou, Čyras, and Toni 2019), potentially exponential AFs arising from structured argumentation and their issues has been discussed, e.g., by Strass, Wyner, and Diller (2019), and infinite arguments for ABA were investigated (Thang, Dung, and Pooksook 2022). In contrast to these works, we relate features of the given ABA instance to the size of the resulting arguments and complexity of reasoning.

We showed that the complexity of credulous acceptance in ABA via instantiating an AF can be confined to either the instantiation step or the AF reasoning step. This is in contrast to the standard instantiation procedure in which both steps are intractable. As we confirm empirically, our results pave the way for efficient instantiation-based ABA reasoning, which previously has not been competitive with directly reasoning on the level of an ABAF. While NP-hardness of, e.g., credulous reasoning is a clear theoretical barrier, an interesting avenue for future research is to combine the reduction of the size and complexity of an instantiated AF, thereby possibly combining strengths of both approaches.

Acknowledgments

This research was funded in whole, or in part, by the Austrian Science Fund (FWF) W1255-N23 and P35632, by the Vienna Science and Technology Fund (WWTF) through project ICT19-065, by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig, and by University of Helsinki Doctoral Programme in Computer Science DoCS. The authors wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

References

Amgoud, L.; Besnard, P.; and Vesic, S. 2014. Equivalence in logic-based argumentation. *J. Appl. Non Class. Logics* 24(3):181–208.

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G. R.; Thimm, M.; and Villata, S. 2017. Towards artificial argumentation. *AI Magazine* 38(3):25–36.

Bao, Z.; Čyras, K.; and Toni, F. 2017. ABAplus: Attack reversal in abstract and structured argumentation with preferences. In *Proc. PRIMA*, volume 10621 of *LNCS*, 420–437. Springer.

Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds. 2018. *Handbook of Formal Argumentation*. College Publications.

Besnard, P., and Hunter, A. 2008. *Elements of Argumentation*. MIT Press.

Bondarenko, A.; Dung, P. M.; Kowalski, R. A.; and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.* 93:63–101.

Craven, R., and Toni, F. 2016. Argument graphs and assumption-based argumentation. *Artif. Intell.* 233:1–59.

Craven, R.; Toni, F.; Cadar, C.; Hadad, A.; and Williams, M. 2012. Efficient argumentation for medical decision-making. In *Proc. KR*, 598–602. AAAI Press.

Čyras, K.; Fan, X.; Schulz, C.; and Toni, F. 2018. Assumption-based argumentation: Disputes, explanations, preferences. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 7, 365–408.

Čyras, K.; Oliveira, T.; Karamlou, A.; and Toni, F. 2021. Assumption-based argumentation with preferences and goals for patient-centric reasoning with interacting clinical guidelines. *Argument Comput.* 12(2):149–189.

Čyras, K.; Heinrich, Q.; and Toni, F. 2021. Computational complexity of flat and generic assumption-based argumentation, with and without probabilities. *Artif. Intell.* 293:103449.

Dimopoulos, Y.; Nebel, B.; and Toni, F. 2002. On the computational complexity of assumption-based argumentation for default reasoning. *Artif. Intell.* 141(1/2):57–78.

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.

Dvořák, W., and Dunne, P. E. 2018. Computational problems in formal argumentation and their complexity. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 14.

Fan, X.; Toni, F.; Mocanu, A.; and Williams, M. 2014. Dialogical two-agent decision making with assumption-based argumentation. In *Proc. AAMAS*, 533–540. IFAAMAS/ACM.

Gaggl, S. A.; Linsbichler, T.; Maratea, M.; and Woltran, S. 2020. Design and results of the second international competition on computational models of argumentation. *Artif. Intell.* 279.

García, A. J., and Simari, G. R. 2004. Defeasible logic programming: An argumentative approach. *Theory Pract. Log. Program.* 4(1-2):95–138.

Karamlou, A.; Čyras, K.; and Toni, F. 2019. Complexity results and algorithms for bipolar argumentation. In *Proc. AAMAS*, 1713–1721. IFAAMAS.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2017. From structured to abstract argumentation: Assumption-based acceptance via AF reasoning. In *Proc. ECSQARU*, volume 10369 of *LNCS*, 57–68. Springer.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2021a. Declarative algorithms and complexity results for assumption-based argumentation. *J. Artif. Intell. Res.* 71:265–318.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2021b. Harnessing incremental answer set solving for reasoning in assumption-based argumentation. *Theory Pract. Log. Program.* 21(6):717–734.

Modgil, S., and Prakken, H. 2013. A general account of argumentation with preferences. *Artif. Intell.* 195:361–397.

Niskanen, A., and Järvisalo, M. 2020. μ-toksia: An efficient abstract argumentation reasoner. In *Proc. KR*, 800–804.

Rapberger, A.; Ulbricht, M.; and Wallner, J. P. 2022. Argumentation frameworks induced by assumption-based argumentation: Relating size and complexity. In *Proc. NMR*, volume 3197 of *CEUR Workshop Proceedings*, 92–103. CEUR-WS.org.

Strass, H.; Wyner, A.; and Diller, M. 2019. *EMIL*: Extracting meaning from inconsistent language: Towards argumentation using a controlled natural language interface. *Int. J. Approx. Reason.* 112:55–84.

Thang, P. M.; Dung, P. M.; and Pooksook, J. 2022. Infinite arguments and semantics of dialectical proof procedures. *Argument Comput.* 13(2):121–157.

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

Yun, B.; Oren, N.; and Croitoru, M. 2020. Efficient construction of structured argumentation systems. In *Proc. COMMA*, volume 326 of *FAIA*, 411–418. IOS Press.

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

A Omitted Proofs of Section 4

Proposition 16. *For each m-rule-size bounded ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ with $|\{r \in \mathcal{R} \mid head(r) = s\}| \leq l$ for all $s \in \mathcal{L}$, there are at most*

$$l^p \cdot |\mathcal{L} \setminus \mathcal{A}|, \quad \textit{with } p = \sum_{i=0}^{k-1} m^i,$$

many tree-based arguments of height $k \geq 1$.

Proof. To prove the statement, we show that the number of all possible trees constructible from D is bounded by $n \cdot l^p$ with $p = \sum_{i=0}^{k-1} m^i$ and $n = |\mathcal{L} \setminus \mathcal{A}|$. Here, we do not require that the leaves of the trees are labeled as assumptions. Observe that the set of all tree-based arguments of height k is a subset of the number of all trees constructible from D.

For each literal $s \in \mathcal{L} \setminus \mathcal{A}$ which appears as head of a rule in D, there are at most $l \cdot x^m$ many trees where x is the maximum number of trees with head c for a literal $c \in body(r)$ for some rule r with $head(s)$. Indeed, there are at most l rules with head s, all bounded by m. We express this correspondence via the function $f(x) = l \cdot x^m$. The total number of trees with root s constructible from D after k steps is thus given by $f^k(1)$. We show that $f^k(1) = l^{\sum_{i=0}^{k-1} m^i}$ via induction over k.

For $k = 1$, we have $f(1) = l$.

Now assume the statement holds true for rule depth $k-1$.

$$\begin{aligned} f^k(1) &= l \cdot (f^{k-1}(1))^m \\ &= l \cdot (l^{\sum_{i=0}^{k-2} k^i})^m \\ &= l \cdot l^{m(\sum_{i=0}^{k-2} m^i)} \\ &= l^{\sum_{i=0}^{k-1} m^i}. \end{aligned}$$

We thus obtain that the number of all possible trees of height k constructible from D is bounded by $l^p \cdot n$ with $p = \sum_{i=0}^{k-1} m^i$. □

Proposition 19. *It is NP-hard to decide whether there is a proof tree from a given set of assumptions to a given claim.*

Proof. Let $\phi = c_1 \wedge \cdots \wedge c_m$ be a Boolean formula in conjunctive normal form over vocabulary $X = \{x_1, \ldots, x_n\}$ with $C = \{c_1, \ldots, c_m\}$ the set of clauses. Construct ABA D with $\mathcal{A} = C$, atoms C together with literals over X and $\{d_{x_1}, \ldots, d_{x_n}\}$, and the following rules (contraries are not relevant). We view clauses as sets of literals. Note that "$\neg x$" is a symbol in ABA and has no meaning attached to the negation sign.

$$\begin{aligned} x &\leftarrow \{c \in C | x \in c\}, \neg x \leftarrow \{c \in C | \neg x \in c\} \text{ for each } x \in X \\ d_x &\leftarrow x, d_x \leftarrow \neg x \text{ for each } x \in X \\ f &\leftarrow d_{x_1}, \ldots, d_{x_n} \end{aligned}$$

We claim that (C, f) is an argument iff ϕ is satisfiable.

First, assume that ϕ is satisfiable and consider a satisfying assignment τ of ϕ. We show that (C, f) is an argument in D. We construct the corresponding tree-based argument as follows: starting from rule $f \leftarrow d_{x_1}, \ldots, d_{x_n}$, we consider rule $d_x \leftarrow x$ in case $\tau(x) = 1$ and rule $d_x \leftarrow \neg x$ in case $\tau(x) = 0$. We show that for each clause $c \in C$, there is at least one leaf labeled c: Consider a clause $c \in C$. Then there is a literal $l \in c$ such that $\tau(l) = 1$. In case $l = x$ for some $x \in X$, we obtain that d_x has a predecessor leaf labeled c (because $x \leftarrow \{c \in C | x \in c\}$ and because we used the rule $d_x \leftarrow x$ in the construction); likewise, in case $l = \neg x$, d_x has a predecessor labeled c. As c was arbitrary, we have shown that (C, f) is an argument in D.

For the other direction, assume (C, f) is an argument in D. Consider the corresponding tree-based argument t. We construct a satisfying assignment of ϕ by inspecting which rules to derive d_x appear in the tree t; that is, we define τ as follows: For an atom $x \in X$, we define $\tau(x) = 1$ if d_x has predecessors $C' \subseteq C$ stemming from chaining rules $d_x \leftarrow x$ and $x \leftarrow C'$; otherwise $\tau(x) = 0$ if d_x has predecessors $C'' \subseteq C$ stemming from chaining the rules $d_x \leftarrow \neg x$ and $\neg x \leftarrow C'$. In case $C' = C''$ we let $\tau(x) = 1$. Observe that τ is well-defined because d_x appears exactly once in the tree-based argument t (otherwise, t is not a tree because it has (at least) two distinctive roots). Thus each atom is assigned either true or false. Moreover, each clause $c \in C$ is satisfied by τ: Given $c \in C$, then there is a literal l and a rule $l \leftarrow C'$ such that $c \in C'$, otherwise t does not contain a leaf labeled c. By construction of D this implies that $l \in c$. In case $l = x$ for some $x \in X$, the tree t is constructed from $d_x \leftarrow x$ and thus $\tau(x) = 1$, consequently c is satisfied; otherwise, $\neg x \in c$ and the tree t is constructed from $d_x \leftarrow \neg x$, thus $\tau(x) = 0$, again, c is satisfied. As c was arbitrary, we have shown that τ is a satisfying assignment of ϕ. □

B Omitted Proofs of Section 5

Proposition 21. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an atomic ABAF. The core F_D of D has $|\mathcal{A}| + |\mathcal{R}|$ many arguments and can be computed in polynomial time.*

Proof. Since D is atomic and flat, each tree-based argument t stems from either an assumption $a \in \mathcal{A}$ yielding the core argument (a, a) or from a rule $s \leftarrow p_1, \ldots, p_n$ yielding the core argument $(\{p_1, \ldots, p_n\}, s)$. No further argument can be constructed. □

Theorem 22. *Credulous reasoning under semantics $\sigma \in \{adm, com, prf, stb\}$ is NP-complete for atomic ABAFs.*

Proof. Consider a CNF ϕ with clauses $C = \{c_1, \ldots, c_m\}$ and Boolean variables $X = \{x_1, \ldots, x_n\}$, let D be the constructed ABA according to Reduction 14, and let $F = (\mathbb{A}, \mathbb{R})$ be the core of D. An example of the construction is given in Figure 2. The ABAF D is atomic, moreover, as discussed in Section 3, the AF F corresponds to the standard translation (Dvořák and Dunne 2018, Reduction 3.6). Hence, ϕ is satisfiable iff the corresponding tree-based argument t with $cl(t) = \phi$ is satisfiable. Due to the semantics correspondence of the ABAF D and its associated AF F, this is the case iff ϕ is credulously accepted in D. □

B.1 Proof of Theorem 27

Towards proving Theorem 27, we observe that the condition 'each $s \in \mathcal{L}$ is contained in $Th(\mathcal{A})$' guarantees that each rule is *applicable*: We say that a rule r is applicable in an ABA framework D if it holds that all body elements of r are derivable in D, or, in different words, there exists a tree-based argument in D whose top rule is r (the one deriving the claim of the tree-based argument).

Since each atom in a rule $r \in \mathcal{R}$ can be derived in an ABA that satisfies that each $s \in \mathcal{L}$ is contained in $Th(\mathcal{A})$ (and thus each proof tree construction terminates) we obtain the following useful lemma.

Lemma 42. *Given an ABA D that satisfies each $s \in \mathcal{L}$ is contained in $Th(\mathcal{A})$. Then each rule is applicable.*

Lemma 43. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be a non-circular ABA framework such that each $s \in \mathcal{L}$ is in $Th_D(\mathcal{A})$ and $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}', ^{-\prime})$ the AF-sensitive ABA framework of D. Moreover, let $E' \in adm(D')$, $E = E' \cap \mathcal{A}$, and $U = \{a \in \mathcal{A} \mid E$ does not attack a in $D\}$. The following statements hold.*

1. *If $x_d \in E'$ then there is a rule $r' \in \mathcal{R}'$ with $head(r') = x$ and $body(r') \subseteq E'$.*
2. *If $x_{nd} \in E'$ then for all rules $r' \in \mathcal{R}'$ with $head(r') = x$ it holds that E' attacks $body(r')$ in D'.*
3. *If $x_d \in E'$ then $x \in Th_{\mathcal{R}}(E)$.*
4. *If $x_{nd} \in E'$ then $x \notin Th_{\mathcal{R}}(E \cup U)$.*

If $E' \in com(D')$ then the following statements hold:

5. *$x_d \in E'$ iff $x \in Th_{\mathcal{R}}(E)$, and*
6. *$x_{nd} \in E'$ iff $x \notin Th_{\mathcal{R}}(E \cup U)$.*

Proof. Let D, D', E, E', and U as defined in the statement of the lemma.

For Item 1, assume that $x_d \in E'$. This implies that there is an $x \in \mathcal{L} \setminus \mathcal{A}$. It holds that $\{x_{nd}\}$ attacks x_d. By admissibility of E', we conclude that E' attacks x_{nd}. Since $\overline{x_{nd}} = x$, it must hold that $x \in Th_{\mathcal{R}'}(E')$. By construction of D', for all $r' \in \mathcal{R}'$ it holds that $body(r') \subseteq \mathcal{A}'$ (only assumptions in bodies). If for each $r' \in \mathcal{R}'$ with $head(r') = x$ we find that $body(r') \nsubseteq E'$ then x is not derivable from E' in D', a contradiction. This implies that there is a rule $r' \in \mathcal{R}'$ with $head(r') = x$ and $body(r') \subseteq E'$ (statement 1 is proven).

For Item 2, assume that $x_{nd} \in E'$ and suppose, for the sake of finding a contradiction, that there is a rule $r' \in \mathcal{R}'$ with $head(r') = x$ such that E' does not attack $body(r')$ in D'. It holds that $body(r')$ attacks x_{nd} in D' (recall that all body elements are assumption and that $\overline{x_{nd}} = x$). If $x_{nd} \in E'$ and E' does not attack $body(r')$ then E' is not admissible in D', a contradiction.

For Item 3, assume that $x_d \in E'$. To show that $x \in Th_{\mathcal{R}}(E)$, we construct a labeled tree $G = (V, A)$ that contains, intuitively speaking, all possible tree-based derivations of x in D. We construct G by iterative construction of different levels (stages) as follows.

- We construct a single node at stage 0 and label it with x_d.
- Assuming G is constructed up to stage i, we construct stage $i+1$ as follows: if v is a vertex constructed in stage i with label $x'_d \in \mathcal{L}'$, then, for each rule r' in D' with $body(r') \subseteq E'$ such that $head(r') = x'$ and $l \in body(r')$, add a vertex y with label $l \in body(r')$ to stage $i+1$, and connect it to v (i.e., add (l, v) to the set of arcs A).

First observe that for each vertex v labeled with x'_d there is always at least one rule $r' \in \mathcal{R}'$ with head x'. This holds because if $x'_d \in \mathcal{A}$ then there is an atom $x' \in \mathcal{L} \setminus \mathcal{A}$ which must be derivable in D, by assumption ($x' \in Th(\mathcal{A})$) and, thus, there is a corresponding rule concluding x' in $\mathcal{R}'$. We show that the process above for construction of G terminates. To see this, suppose otherwise. Then there is an infinite sequence of vertices $(v_1, v_2, \ldots)$ starting from a vertex v_1 labeled with x_d and an between neighboring vertices in the sequence. Since there are finitely many labels, there are two vertices with the same label in the sequence. Let these vertices be v_i and v_j, $i < j$. Then one can construct a derivation tree in D s.t. there is a path from an assumption in this tree to the root visiting the same label of v_i and v_j twice: by derivability of all atoms, we can find a derivation of the label of v_j, and extend it to v_i through the rules corresponding to the rules required to make up the sequence. This contradicts the presumption that D is non-circular.

By presumption that D is non-circular, we conclude that G is a finite tree, with root x_d. The leaves of this tree are assumptions in $\mathcal{A}$ (in fact in E): suppose a leaf is labeled by some y_d. Then, by admissibility of E', we find that there is some rule $r' \in \mathcal{R}'$ with $body(r') \subseteq E'$ and $head(r') = y$ (Item 1). But then this cannot be a leaf. Together with termination, we conclude that leaves must be assumptions in D not of the form z_d for some $z \in \mathcal{L} \setminus \mathcal{A}$ (i.e., leaves are labeled by assumptions in $\mathcal{A}$ and by the previous reasoning and construction also in E). Then G contains a subtree that directly corresponds to a derivation of x in D by E (leaves are assumptions in $\mathcal{A}$ and each internal node can be labeled according to derivation trees). This implies that $x \in Th_{\mathcal{R}}(E)$.

For Item 4, assume that $x_{nd} \in E'$. Suppose that $x \in Th_D(E \cup U)$. Then there is a tree-based argument $E \cup U \vdash_{\mathcal{R}} x$ in D. By Item 2, whenever $y_{nd} \in E'$ then E' attacks $body(r')$ for all rules $r' \in \mathcal{R}'$ with $head(r') = x$. Iteratively go through the tree-based argument, starting with $s^1 = x$ and $i = 1$. We find that there must be a rule $r \in \mathcal{R}$ that is part of the tree-based argument with $head(r) = s^i$ and, thus, a rule $r' \in \mathcal{R}'$ with $head(r') = s^i$. Moreover, $s^i_{nd} \in E'$. We already showed that E' attacks $body(r')$ on some $w \in body(r')$. Consider two cases.

- $w \in \mathcal{A}$. Then $w \in E \cup U$ (since $E \cup U \vdash_{\mathcal{R}} x$ is the tree-based argument). It holds that E' derives $\overline{w}$ in D'. If $\overline{w} \in \mathcal{A}$ then $\overline{w} \in E'$. If $w \in E$ then E' is not conflict-free: $w \in E'$ and $\overline{w}$ is derivable from E' in D', a contradiction. If $w \in U$, then (i) either $\overline{w} \in \mathcal{A}$ and E attacks U in D (a contradiction) or (ii) $\overline{w} \notin \mathcal{A}$ and $\overline{w} = y_d$ with $y \in \mathcal{L} \setminus \mathcal{A}$. Then $y \in Th_{\mathcal{R}}(E)$, contradicting that E does not attack U in D.
- $w \notin \mathcal{A}$. Then $w = y_d$ with $y \in \mathcal{L} \setminus \mathcal{A}$. By construction, $y_{nd} \in E'$ and there is a rule $r \in \mathcal{R}$ part of the tree-based

argument concluding y. By Item 3 we again infer that E' attacks $body(r')$ of the corresponding rule $r' \in \mathcal{R}$. We set $s^{i+1} = y$ and continue.

This process terminates, as a tree-based argument is assumed to be finite. This implies a contradiction of the form as in the previous first item: E attacks either E or U. Thus, we conclude that $x \notin Th_D(E \cup U)$.

For Item 5, assume that $x \in Th_{\mathcal{R}}(E)$. Let $T = E \vdash_{\mathcal{R}} x$ be a tree-based argument that witnesses the derivation. We show by induction on the height of the derivation that if there is a node v labeled l in (a derivation tree of) T then then (i) $l \in E'$ and E' defends l in D' or (ii) $l \notin E'$ and E' defends l_d in D'. For the base case (height 0) we find that $l \in E$. The statement holds directly ($E = E' \cap \mathcal{A}$) and by assumption that E' is complete in D'. Assume that the statement holds up to height i. Then $l \in \mathcal{L} \setminus \mathcal{A}$ (D is flat) and there is a rule $r \in \mathcal{R}$ with $head(r) = l$ and the children of the current node are $body(r)$. Since E' is complete, and by assumption that the statement holds up to i, we infer that E' defends $body(r')$ of the corresponding rule $r' \in \mathcal{R}'$ to r. But then $body(r') \subseteq E'$. Then E' attacks l_{nd} (E' derives l). Then E' defends l_d.

For Item 6, assume that $x \notin Th_{\mathcal{R}}(E \cup U)$. First we prove that E' attacks any $a \in \mathcal{A} \setminus (E \cup U)$ in D'. Suppose E' does not attack such an a in D'. Then $\overline{a} \notin Th_{\mathcal{R}'}(E')$. If $\overline{a}$ is an assumption in D', then $\overline{a} \in \mathcal{A}$ (by construction of D'), and E does not attack a in D. But then $a \in U$. If $\overline{a}$ is not an assumption in D' then $\overline{a} = y$ and there is an $y_d \in \mathcal{A}'$. We infer that y_d is not in E' (since E' does not derive y in D' and then y_d is not defended by E' in D'). By Item 5, we conclude that $Th_{\mathcal{R}}(E)$ does not contain y. But then E does not attack a in D, a contradiction to $a \notin E \cup U$.

Suppose for the sake of finding a contrary that $x_{nd} \notin E'$. Then there is an $X \subseteq \mathcal{A}'$ that attacks x_{nd} and E' does not attack X (both in D'). If $X \subseteq \mathcal{A}$, then $X \subseteq E \cup U$ (by reasoning above, E' attacks all assumptions in $\mathcal{A}$ outside of $E \cup U$ in D'). But then $x \in Th_{\mathcal{R}}(E \cup U)$, a contradiction. Then $X \nsubseteq \mathcal{A}$ and X contains some y_d. This y_d is not attacked by E' in D', i.e., $y_{nd} \notin E'$. By the same reasoning above, we can find a rule in D' whose body contains a z_d not attacked by E'. Continuing this iteratively, we first terminate (no acyclic derivations) and reach rules with bodies only in $\mathcal{A}$. By the above, we arrive at a contradiction. □

We are now ready to prove the theorem.

Theorem 27. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be a non-circular ABA framework such that each $s \in \mathcal{L}$ is in $Th_D(\mathcal{A})$ and D' the AF-sensitive ABA framework of D. It holds that*

- *if $E \in adm(D)$, then there is an $E' \in adm(D')$ with $E = E' \cap \mathcal{A}$ and $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E') \cap \mathcal{L}$, and*
- *if $E' \in adm(D')$, then $E' \cap \mathcal{A} \in adm(D)$ and $Th_{\mathcal{R}}(E) \supseteq Th_{\mathcal{R}'}(E') \cap \mathcal{L}$.*

Moreover, for $\sigma \in \{com, prf, stb\}$ we find that

- *if $E \in \sigma(D)$, then there is an $E' \in \sigma(D')$ with $E = E' \cap \mathcal{A}$ and $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E') \cap \mathcal{L}$, and*
- *if $E' \in \sigma(D')$, then $E' \cap \mathcal{A} \in \sigma(D)$ and $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E') \cap \mathcal{L}$.*

Proof. Let D and D' be given as stated.

($\Rightarrow$) First, let $E \in adm(D)$. Construct

$$U = \{u \in \mathcal{A} \mid E \text{ does not attack } u \text{ in } D\},$$
$$B = \{s \in \mathcal{L} \setminus \mathcal{A} \mid s \notin Th_{\mathcal{R}}(E \cup U)\}.$$

The intuition is that U contains "undefeated" assumptions which are not attacked by E in D, and B contains all non-assumptions whose derivation is "blocked" by E in the sense that to derive them one requires an attacked assumption. Now we let

$$E' = E \cup \{s_d \mid s \in Th_{\mathcal{R}}(E) \setminus \mathcal{A}\} \cup \{s_{nd} \mid s \in B\} \quad (1)$$

That is, E' contains all assumptions in E, all assumptions s_d whenever s is derivable from E in D, and all assumptions s_{nd} whenever the derivation of s is "blocked" in D.

Our first step is to show that $E' \in cf(D')$. Suppose the contrary, i.e. $E' \notin cf(D')$. Then there is an $a' \in E'$ s.t. $\overline{a'} \in Th_{\mathcal{R}'}(E')$. We consider three cases:

(Case 1) a' is of the form s_d for some $s \in \mathcal{L} \setminus \mathcal{A}$: We infer that $s_d \in E'$ and $s_{nd} \in E'$ (only the latter assumption attacks the former). However, by (1), $s_d \in E'$ implies $s \in Th_{\mathcal{R}}(E)$ and $s_{nd} \in E'$ implies $s \in B$, i.e. $s \notin Th_{\mathcal{R}}(E \cup U)$. This is a contradiction since derivability is upward-closed under $\subseteq$.

(Case 2) a' is of the form s_{nd} for some $s \in \mathcal{L} \setminus \mathcal{A}$: Hence it holds that $s_{nd} \in E'$ and there is a rule $r' \in \mathcal{R}'$ s.t. $head(r') = \overline{s_{nd}} = s$ and $body(r') \subseteq E'$ (all rules in D' contain only assumptions). Observe that in D' no assumption of the form s_{nd} occurs in the body of any rule, i.e., we even have $body(r') \subseteq E \cup \{s_d \mid s \in Th_{\mathcal{R}}(E)\}$. Now s is derivable from E in D. To see this, note that there is a corresponding rule r to r' in D, and for each body element in r it holds that E either contains the body element, if it is an assumption, or derives the body element. We arrive at a contradiction: $s_{nd} \in E'$ implies $s \in B$, in turn implying that s is not derivable from E in D.

(Case 3) $a' \in \mathcal{A}$: Then there is a rule $r' \in \mathcal{R}'$ with $head(r') = \overline{a'}$, and $body(r') \subseteq E'$. By construction, it follows that there is a rule $r \in \mathcal{R}$ with $head(r) = head(r')$ and the body elements of r and r' are the same, except for non-assumptions $s \in body(r)$ which are replaced by s_d in r'. By construction of E', it follows directly that $body(r) \subseteq Th_{\mathcal{R}}(E)$ (since $E \subseteq E'$ and all derivable atoms from E are present as s_d in E', implying that for all body elements s_d in $body(r)$ one has a body element s in $body(r)$ and one can derive s from E). This implies that $E \notin cf(D)$, a contradiction.

We conclude $E' \in cf(D')$. Now suppose that $E' \notin adm(D')$. This implies that there is an assumption set $A' \subseteq \mathcal{A}'$ s.t. A' attacks E' in D', but E' does not attack A' in D'. By definition of attack, it holds that A' attacks E' on some $a' \in E'$ and $\overline{a'} \in Th_{\mathcal{R}'}(A')$. Consider again three cases.

(Case 1) a' is of the form s_{nd} for some $s \in \mathcal{L} \setminus \mathcal{A}$: Since s is not an assumption in D', it holds that A' derives s with a rule $r' \in \mathcal{R}'$ s.t. $head(r') = s$ (the contrary of s_{nd} in D') and $body(r') \subseteq \mathcal{A}'$ implies $body(r') \subseteq A'$ (no body element can be derived). Since each rule is applicable in D, there exists a tree-based argument α in D with $cl(\alpha) = s$ and its top rule being r, the corresponding rule to r' in D. By construction, it holds that $body(r) \cap B \neq \emptyset$ or E attacks an assumption in $body(r)$, because if $s_{nd} \in E'$, then $s \in B$. If E does not attack any assumption in $body(r)$ and $body(r)$ does not contain any element of B, it holds that one can derive all body elements from $E \cup U$ in D. If there is a body element s' of $body(r)$ in B, then E' attacks A': the corresponding s'_{nd} is in E', and s'_d in A'. If E attacks an assumption in $body(r)$, then there is a tree-based argument starting from E that claims the contrary of this assumption. Then E' derives this contrary, as well (again the top rule of the tree-based argument has a corresponding rule in D', and all its body elements are in E'). We arrive at a contradiction: E' does attack A' in D'.

(Case 2) a' is of the form s_d for some $s \in \mathcal{L} \setminus \mathcal{A}$: This implies that $s_{nd} \in A'$ (only this assumption attacks s_d). By construction of E', it holds that $s \in Th(E)$. This implies that there is a tree-based argument $A* \vdash s$ for some $A* \subseteq E$. Consider the top rule r of this tree-based argument. It holds that $head(r) = s$, and all elements of $body(r)$ are derivable from E. This implies that there is a corresponding rule $r' \in \mathcal{R}'$ with the same head and body element modified as stated in the construction of D'. It follows that $body(r') \subseteq E'$ (all assumptions in $body(r)$ are in E' and all remaining atoms are derivable from E, for which we showed that the corresponding assumption is in E'). But then $s \in Th_{\mathcal{R}'}(E')$ and $\overline{s_{nd}} = s$, implying that E' attacks A', again contradicting non-admissibility.

(Case 3) $a' \in \mathcal{A}$: Then there exists a rule r' deriving from A' the contrary of a', and a tree-based argument with corresponding rule r as its top rule. If E' does not attack A' in D', it holds that from E' one does not derive the contrary of any assumption in $\mathcal{A}$ in $body(r')$ and for all s_d in $body(r')$, E' does not contain s_{nd}. This implies that the contrary of a' is derivable from $E \cup U$ (no atom in $body(r)$ is blocked or a contrary derivable from E by construction of E'). But then E is not admissible in D: $a' \in E' \cap \mathcal{A} = E$, thus $a' \in E$ and $E \cup U$ derives $\overline{a'}$ in D, a contradiction to the presumption that E is admissible.

Now consider some $s \in \mathcal{L} \setminus \mathcal{A}$. It holds that $s \in Th_{\mathcal{R}}(E)$ iff there is a tree-based argument $(A' \vdash s)$ in D' with $A' \subseteq E$ iff there is a rule $r' \in \mathcal{R}'$ with $s = head(r')$ and $body(r') \subseteq E'$ iff $s \in Th_{\mathcal{R}'}(E')$. This concludes the first item to be proven.

($\Leftarrow$) Now, assume that $E' \in adm(D')$. Construct $E = E' \cap \mathcal{A}$. We first show that if $s_d \in E'$, for some $s \in \mathcal{L} \setminus \mathcal{A}$, then $s \in Th_{\mathcal{R}}(E)$ (i.e., if s_d is part of E', then s is derivable from E in D). This holds by Lemma 43. Now assume that $s \in Th_{\mathcal{R}'}(E')$. Then there is a rule $r' \in \mathcal{R}'$ s.t. $body(r') \subseteq E'$ and $head(r') = s$. Let r be the corresponding rule in D. It holds that E derives (or contains) all body elements of r: if a body element is an assumption in $\mathcal{A}$ then both E and E' contain the assumption, if a body element x is not an assumption in $\mathcal{A}$ then $x_d \in E'$ and, by the lemma, we infer that E derives x in D. Thus, if E' derives s in D' it holds that E derives s in D (showing part of the claim in the second item).

Now, suppose $E \notin cf(D)$. Then there is an $a \in E$ and $\overline{a} \in Th_{\mathcal{R}}(E)$. If $\overline{a} \in E$, then E' is not conflict-free in D', a contradiction. So $\overline{a} \notin E$ and is not an assumption in $\mathcal{A}$. Say $\overline{a} = s$ for some $s \in \mathcal{L} \setminus \mathcal{A}$. We show that E' derives s: as $s \in Th_{\mathcal{R}}(E)$, there is a tree-based argument $A \vdash s$ with $A \subseteq E$ with top-rule r. Consider the corresponding rule r' in D'. By admissibility of E' it holds that E' attacks $body(r')$ (to defend itself against the attack on $a \in E'$). Now, if E' attacks some $b \in body(r') \cap \mathcal{A}$, then E' is not conflict-free, as $body(r') \cap \mathcal{A} \subseteq E'$ (recall that r can contain as assumptions only elements in $E \subseteq E'$). Thus E' attacks some $x_d \in body(r')$. Thus E' contains x_{nd}. By Lemma 43, it follows that $x \notin Th_{\mathcal{R}}(E)$, a contradiction ($E$ does not derive x in D).

Suppose that $E \notin adm(D)$. The proof proceeds analogous to proving conflict-freeness of E: If $E \notin adm(D)$, then there is an assumption set $A \subseteq E \cup U$ that attacks E. Then A derives a contrary s of an assumption in E, thus there is a tree-based argument $A' \vdash s$ with $A' \subseteq A$ and top-rule r in D. We consider the corresponding rule r' in D'. By admissibility of E', it holds that E' attacks $body(r')$ on some $a' \in body(r')$. This implies that E' derives $\overline{a'}$ in D'. Consider two cases: $a' \in \mathcal{A}$ or $a' \notin \mathcal{A}$ (is an original assumption or not). In the first case, $\overline{a'} = x$ and either $x \in \mathcal{A}$ (then E directly attacks A, a contradiction) or $x \notin \mathcal{A}$. Then there is a rule in D' deriving x and all the body elements are in E'. But then E derives x, as well (by Lemma 43 all non-assumptions in the body are derivable by E; contradicting again our presumption). If a' is not in $\mathcal{A}$, then $a' = x_d$ for some $x \in \mathcal{L} \setminus \mathcal{A}$ and $\overline{a'} = x_{nd}$. Then $x \notin Th_{\mathcal{R}}(E \cup U)$, implying that E attacks A' (since A' must contain some assumption attacked by E).

For complete semantics, let $\sigma = com$. Assume that $E \in com(D)$ and construct E' as above for the admissible case. By the results above, we find that $E' \in adm(D')$ (since $E \in adm(D)$). Suppose, for the sake of inferring a contradiction, that $E' \notin com(D')$. That is, E' is admissible but not complete in D'. This implies that there is an assumption set $A' \subseteq \mathcal{A}'$ such that whenever there is a $B' \subseteq \mathcal{A}'$ attacking A' we find that E' attacks B', but $A' \nsubseteq E'$ (E' defends A' but does not contain A'). W.l.o.g. we can assume that $A' = \{a'\}$ is a singleton (if E' defends A' then E' defends any $a' \in A'$). As before, we consider three cases.

(Case 1) a' is of the form s_d for some $s \in \mathcal{L} \setminus \mathcal{A}$. Then $B = \{s_{nd}\}$ attacks a' (and all sets attacking a' contain s_{nd} by construction). By assumption, E' attacks B, and E' derives s in D' (by construction s is the contrary of s_{nd}). Then there is a rule $r' \in \mathcal{R}'$ such that $head(r') = s$ and $body(r') \subseteq E'$. There is a corresponding rule $r \in \mathcal{R}$ in D. By construction $E \subseteq E'$. For each non-assumption $x \in body(r)$ we find that $x_d \in E'$. By construction it holds that $x \in Th_{\mathcal{R}}(E)$. Then $s \in Th_{\mathcal{R}}(E)$. By construction, $s \in Th_{\mathcal{R}}(E)$ implies that $s_d \in E'$, a contradiction.

(Case 2) a' is of the form s_{nd} for some $s \in \mathcal{L} \setminus \mathcal{A}$. Then s_{nd} is attacked by each $r' \in \mathcal{R}'$ with $head(r') = s$. Consequently, E' attacks then each $body(r') \subseteq \mathcal{A}'$. If $s \notin Th_{\mathcal{R}}(E \cup U)$ we find that $s \in B$ and, by construction, we have $s_{nd} \in E'$. Because, by assumption, we have $s_{nd} \notin E'$ it must be that $s \in Th_{\mathcal{R}}(E \cup U)$. Consider $X \subseteq E \cup U$ such that there is a tree-based argument with

$X \vdash s$ in D. Let r be the topmost rule in this argument. We have $head(r) = s$. By reasoning above, we find that for the corresponding rule $r' \in \mathcal{R}'$ it holds that E' attacks $body(r')$ on some assumption $b' \in body(r')$. This implies that $\overline{b'}$ is derivable from E' in D' via some rule in D'. Then E derives $\overline{b'}$, as well: for a corresponding rule all body elements are either in E or derivable from E (by construction of E'). Then E attacks b', contradicting that X is not attacked by E in D. If $b' \notin \mathcal{A}$ then $b' = y_d$ for some $y \in \mathcal{L} \setminus \mathcal{A}$. By reasoning above and E' attacks $body(r')$ we find that $y_{nd} \in E'$. By construction we infer that $y \notin Th_{\mathcal{R}}(E \cup U)$, a contradiction (the body of r in the argument above is not derived via $E \cup U$, implying that the argument is not an argument).

(Case 3) $a' \in \mathcal{A}$. Then E' attacks each $body(r')$ for $r' \in \mathcal{R}'$ whenever $head(r') = s = \overline{a'}$. By construction, also E does not contain a'. Consider any $A \vdash s$ in D. The body of rule r' corresponding to the topmost rule in this tree-based argument is attacked by E'. If attacked on an assumption in $\mathcal{A}$ then E attacks A. Otherwise, E' attacks on some x_d via containing x_{nd}. By construction we find that $x \notin Th_{\mathcal{R}}(E \cup U)$, and E attacks A.

For the other direction, assume that $E' \in com(D')$ and $E \notin com(D)$. By the results above, $E \in adm(D)$. This implies that there is some $a \in \mathcal{A} \setminus E$ s.t. whenever some $B \subseteq \mathcal{A}$ attacks $\{a\}$ we find that E attacks B.

Consider any $B' \subseteq \mathcal{A}'$ that attacks $\{a\}$ in D'. Such a set B' exists, because there is some $B \vdash \overline{a}$ in D, and the top rule r. The corresponding top rule r' with $body(r') = B'$ and $head(r') = \overline{a}$. For any $x_d \in B'$ we find that E' does not contain x_{nd} (E' does not attack B), implying, by Lemma 43, that $x \in Th_{\mathcal{R}}(E \cup U)$. For any $b \in B' \cap \mathcal{A}$ it holds that E' does not attack b and does not derive $\overline{b}$ in D'. This implies that E does not derive $\overline{b}$, either, since assuming otherwise, by Lemma 43, we find that either $\overline{b} \in E'$ (if $\overline{b} \in E$) or $\overline{b}_d \in E'$ and, because E' is admissible, E' derives $\overline{b}$ (to defend against against $\overline{b}_{nd}$). This leads to a contradiction: E does not attack a tree-based argument for $\overline{a}$ ($E \cup U$ derives $\overline{a}$ in D).

By Lemma 43 we find that $x \in Th_{\mathcal{R}}(E)$ implies $x_d \in E'$. Then $x \in Th_{\mathcal{R}'}(E')$ (otherwise x_{nd} is not attacked by E' and E' not admissible). This implies the claim for complete semantics.

Assume that $E \in stb(D)$ and construct E' as above for admissible semantics. Suppose that $E' \notin stb(D')$. We find that E' is complete in D' (by reasoning above). This implies that there is some $a' \in \mathcal{A}' \setminus E'$ that E' does not attack in D'. As above, consider three cases.

(Case 1) a' is of the form s_d for some $s \in \mathcal{L} \setminus \mathcal{A}$. Then E' does not contain s_{nd}. Then, by construction, $s \notin B$ and $s \in Th_{\mathcal{R}}(E \cup U)$. By assumption, $s_d \notin E'$, implying that $s \notin Th_{\mathcal{R}}(E)$. But then there is a tree-based argument for s containing assumptions from $E \cup U$ and at least one from U that is not in E. This is a contradiction: U must be a subset of E (stable assumption sets contain all unattacked assumptions).

(Case 2) a' is of the form s_{nd} for some $s \in \mathcal{L} \setminus \mathcal{A}$. Then E' does not contain s_d, since $\{s_{nd}\}$ attacks then E' and E' must attack back (by admissibility). The same reasoning as for Case 1 applies.

(Case 3) $a' \in \mathcal{A}$. Then E attacks a' in D. With similar reasoning as in several cases above, we arrive at the fact that E' attacks a', since some tree-based argument with base in E must conclude a contrary of a', and by construction of E' we find that E' derives the contrary, as well.

For the other direction, assume that E' is stable in D' and $E = E' \cap \mathcal{A}$ is not stable in D (but complete, by reasoning above). This means some $a \in \mathcal{A} \setminus E$ is not attacked by E. It holds that a is attacked by E' in D'. By Lemma 43, we find that for some rule r' that concludes the contrary of a, all body elements are in E' and that E derives the contrary, as well.

Assume that $E \in prf(D)$ and that E', constructed as for the admissible case, is not preferred in D'. By reasoning above, we find that E' is complete in D'. This implies that there is some $E'_1 \supset E'$ and E'_1 is complete in D'. Then $E'_1 \cap \mathcal{A}$ is admissible in D. This implies that $E'_1 \setminus E'$ contains only assumptions not in $\mathcal{A}$ (otherwise E would not be preferred in D). If $s_d \in E'_1 \setminus E'$, then $s \notin Th_{\mathcal{R}}(E)$ (by construction) and, since $E = E'_1 \cap \mathcal{A}$ and due to Lemma 43, we find that $s \in Th_{\mathcal{R}}(E)$, a contradiction. If $s_{nd} \in E'_1 \setminus E'$, then, similarly, we conclude both $s \in Th_{\mathcal{R}}(E \cup U)$ and $s \notin Th_{\mathcal{R}}(E \cup U)$, by Lemma 43.

Assume that $E' \in prf(D')$ and that $E = E' \cap \mathcal{A}$ is not preferred in D (but complete, by statements above). Then there is an $E_1 \supset E$ that is complete in D. Then E'_1 constructed as for the admissible case above from E_1 is complete in D'. Since there is some $a \in E_1 \setminus E$ we find that $a \in E'_1 \setminus E'$, a contradiction to E' being preferred in D'. □

B.2 Omitted proofs of Section 5.3

Lemma 29. *Let D be an ABAF and $F = (\mathbb{A}, \mathbb{R})$ the core of D. Let $x = (A, s) \in \mathbb{A}$ be a redundant argument. Then for each $\sigma \in \{adm, com, prf, stb\}$ we have*

$$\{cl(E) \mid E \in \sigma(F)\} = \{cl(E) \mid E \in \sigma(F{\downarrow_{\mathbb{A}\setminus\{x\}}})\}.$$

Proof. Let $x = (A, s)$ and $x^* = (A^*, s)$ be the reason for x to be redundant, i.e., $A^* \subseteq A$. Suppose $x \neq x^*$. Let us start with admissible semantics.

($\subseteq$) Suppose $E \in adm(F)$.

i) First assume $x \in E$. Then E also defends x^* and hence $E^* = E \cup \{x^*\} \setminus \{x\} \in adm(F)$ as well with $cl(E) = cl(E^*)$. Clearly, $E^* \in adm(F{\downarrow_{\mathbb{A}\setminus\{x\}}})$ so we found the corresponding extension in $F{\downarrow_{\mathbb{A}\setminus\{x\}}}$.

ii) Now suppose $x \notin E$. Then it also holds that $E \in adm(F{\downarrow_{\mathbb{A}\setminus\{x\}}})$.

In both cases i) and ii) we manage to keep an extension with the same conclusions in $F{\downarrow_{\mathbb{A}\setminus\{x\}}}$.

($\Leftarrow$) Let $E \in adm(F{\downarrow_{\mathbb{A}\setminus\{x\}}})$.

i) If $x^* \notin E$, then either E attacks x^* or x^* does not attack E. Both cases would then also be true for x –if E attacks x^*, then E attacks x as well– so $E \in adm(F)$ holds as well.

ii) if $x^* \in E$, then $E \in adm(F)$ can be seen (x is certainly no threat to the extension).

Again, the accepted claims can be transitioned from $F{\downarrow_{\mathbb{A}\setminus\{x\}}}$ to F.

For the other semantics we reason analogously. □

Proposition 30. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA framework and $F = (\mathbb{A}, \mathbb{R})$ the core of D. If $(A, s) \in \mathbb{A}$ is not redundant, then there is some tree-based argument t with $leaves(t) = A$ and $cl(t) = s$ in D such that the derivation depth of t is at most $|\mathcal{L} \setminus \mathcal{A}|$.*

Proof. Suppose we are given a tree-based argument $A \vdash s$ with depth greater than $|\mathcal{L} \setminus \mathcal{A}| + 1$. Since D is flat, only one assumption can occur on each path in $A \vdash s$. Hence in any path whose length is greater than $|\mathcal{L} \setminus \mathcal{A}| + 1$ there must be some atom $p \in \mathcal{L}$ occurring twice. Hence $A \vdash s$ contains two sub-arguments $B \vdash p$ and $B' \vdash p$ where $B \vdash p$ is a sub-argument of $B' \vdash p$; that is, we have $B \subseteq B'$.

By replacing $B' \vdash p$ with $B \vdash p$ in the derivation tree $A \vdash s$ we find a tree based argument $\big(A \setminus (B' \setminus B)\big) \vdash s$. If $B \subsetneq B'$, then $A \vdash s$ is redundant; hence it must hold that $B = B'$, and therefore the two core arguments (A, s) and $(A \setminus (B' \setminus B), s)$ coincide.

We proceed like this for each path in $A \vdash s$ that is longer than $|\mathcal{L} \setminus \mathcal{A}| + 1$. This way, we find an argument $A^* \vdash s$ s.t.

- the tree-based argument $A^* \vdash s$ has derivation depth at most $|\mathcal{L} \setminus \mathcal{A}| + 1$, and
- $A = A^*$, i.e., both tree-based arguments represent the same core argument. ☐

Lemma 33. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABAF. If there is a non-redundant argument t in D then there is also an argument t' in D° with $leaves(t') = leaves(t)$ and $cl(t') = cl(t)$. Vice versa, if t is an argument in D°, then there is also an argument t' in D satisfying the same conditions.*

Proof. ($\Rightarrow$) As in the construction of D° we let $k = |\mathcal{L} \setminus \mathcal{A}|$.

We show the claim for each tree-based argument t (with leaves A and root s) with derivation depth $l \leq k$; thereby we proceed per induction over l. More precisely, or inductive hypothesis is as follows:

> If t with $leaves(t) = A$ and $cl(t) = s$ has derivation depth l in D, then t^j with leaves $leaves(t^j) = A$ and root $cl(t^j) = s^j$ is constructible in D° for each $l \leq j \leq k$

Let $l = 1$. Hence in D we can derive s via a rule

$$r = s \leftarrow p_1, \dots, p_n$$

s.t. $p_1, \dots, p_n \in \mathcal{A}$. The same rule occurs in D° and we are done.

Now suppose the claim holds for each $l < k$ and consider some argument t with

$$leaves(t) = A \qquad\qquad cl(t) = s$$

constructible in D with derivation depth $l+1 \leq k$. Consider the top-most rule r which is used to derive s. This rule is of the form $s \leftarrow p_1, \dots, p_n$. By definition, there are sub-arguments $t_1, \dots, t_n$ with

$$cl(t_i) = p_i \qquad \bigcup leaves(t_i) \cup (\{p_1, \dots p_n\} \cap \mathcal{A}) = A$$

and again by definition their derivation depth at most l.

By the inductive hypothesis, in D° we have arguments t_i^j for each $1 \leq i \leq t$ and each $l \leq j \leq k$, i.e.,

$$cl(t_i^j) = p_i^j$$

By construction of D° and due to the rule $s \leftarrow p_1, \dots, p_t$, there is for each $l \leq j \leq k$ a rule with head s^{j+1} and $\{p_1^j, \dots p_t^j\}$ as body. Using the sub-arguments t_i^j we found above, we can apply this rule in order to obtain the desired argument t^{j+1} with $leaves(t^{j+1}) = A$ and $cl(t^{j+1}) = s^{j+1}$.

($\Leftarrow$) This direction is straightforward: In a tree-based argument t in D° replace each rule applied rule

$$s^{j+1} \leftarrow p_1^j, \dots p_n^j$$

with the original one

$$s \leftarrow p_1, \dots p_n$$

and suitably adjust the atoms, i.e., replace each s^{j+1} with s and p_i^j with p_i. This way, a valid tree-based argument in D is obtained. ☐

Corollary 34. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA. Then D° is non-circular and satisfies $\sigma(D) = \sigma(D^\circ)$ for each $\sigma \in \{adm, com, prf, grd, stb\}$.*

Proof. Non-circularity is by construction. Due to Corollary 46 D° contains all non-redundant arguments and application of Lemma 29 yields $\sigma(D) = \sigma(D^\circ)$. ☐

C Omitted proofs of Section 6

Proposition 39. *Credulous reasoning under semantics $\sigma \in \{adm, com, prf, stb\}$ is* NP*-complete for symmetric ABAFs.*

Proof. This is immediate from Reduction 37 since here we construct a symmetric ABAF s.t. ϕ is credulously accepted (under all mentioned semantics) iff the input CNF-formula is satisfiable. ☐

Proposition 40. *Credulous reasoning under semantics $\sigma \in \{adm, com, prf, stb\}$ is decidable in polynomial time in cores of symmetric ABAs.*

Proof. Constructing the cores results in AFs with $|\mathcal{A}|/2$ many even cycles of length 2 satisfying that all arguments with claim $s \notin \mathcal{A}$ have only incoming attacks. Credulous reasoning under admissibility in such AFs is decidable in time polynomial in the number of arguments, since it suffices to check if there exists an argument having the queried claim that is not attacked by both arguments in a 2-cycle.

As before, the result transfers to complete and preferred semantics.

For stable semantics, we first observe that all self-attacking arguments in the corresponding AF (induced by assumptions) are singletons. Indeed, given an assumption $a \in \mathcal{A}$ with $\overline{a} = a$, and consider an argument $b \in \mathcal{A}$ with $\overline{b} = a$. By symmetry, we have $\overline{a} = b$ hence $b = a$. To decide credulous acceptability w.r.t. stable semantics we first check whether a stable extension exists by checking whether a self-attacking argument exists. If yes, we are done (the answer is negative). If not, it holds that preferred and stable semantics coincide. We proceed as before in the case of deciding credulous acceptance for admissible semantics. □

Theorem 41. *Let $\sigma \in \{adm, com, prf, stb\}$. For each ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ and $p \in \mathcal{L}$ one can construct an ABA framework D' in polynomial time s.t. (1) x is credulously accepted in D' iff x is credulously accepted in D w.r.t. σ, and (2) credulous acceptance is in P under σ in the corresponding AF F.*

Proof. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework, and $a \in \mathcal{L}$. It holds that credulous reasoning under admissible, complete, preferred, and stable semantics in the considered ABA fragment is in NP (Bondarenko et al. 1997; Dvořák and Dunne 2018). By NP-completeness of the Boolean satisfiability problem, it follows that there is a Boolean formula ϕ s.t. ϕ is satisfiable iff a is credulously accepted under admissibility in D. By applying Reduction 37 (see also proof of Theorem 22 for proving the reduction) one can reduce ϕ, in polynomial time, to an ABA framework $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}', ^{-'})$ and $a' \in \mathcal{L}'$ s.t. ϕ is satisfiable iff a' is credulously accepted in D'. By construction, D' is symmetric. One can uniformly rename atoms in D' s.t. a' is named a. □

D Extending Corollary 34 to SCCs (Correctness of Implementation)

We can extend Corollary 34 to strongly connected components (SCCs) in the ABA framework as follows.

Definition 44. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework. The* dependency graph *for D is defined as $\mathcal{G}_D = (V, E)$ where $V = \mathcal{L}$ and $(p, q) \in E$ iff there is some rule $r \in \mathcal{R}$ s.t. $p \in body(r)$ and $q = head(r)$. We say $S \subseteq \mathcal{L}$ is an SCC iff S corresponds to some SCC in $\mathcal{G}_D$.*

Observe that each assumption induces its own trivial SCC since our ABA frameworks are flat. By definition, atoms in each SCC are connected in the path of tree-derivations in the following sense.

Lemma 45. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework and $S \subseteq \mathcal{L}$ an SCC. Let $p_1, \dots, p_n$ be a path in a tree-based argument $A \vdash p$. If $p_i \in S$ and $p_j \in S$, then $p_l \in S$ for each $i \le l \le j$.*

Proof. By definition, p_l is reachable from p_i and since p_i is reachable form p_j, it is reachable from p_l as well. We deduce $p_l \in S$. □

For each path P in a derivation tree, let us denote by $P(S)$ the (unique and connected) sub-path whose atoms occur in S. We can extend the claim from Proposition 30 to each single SCC (by the structure of SCCs with the same proof).

Corollary 46. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework and $F = (\mathbb{A}, \mathbb{R})$ the core of D. If $(A, s) \in \mathbb{A}$ is not redundant, then there is some tree-based argument $A \vdash s$ in D such that for each path P in the derivation tree and each SCC S, it holds that $|P(S)| \le |S|$.*

If $A \vdash s$ satisfies the properties from Corollary 46 we say $A \vdash s$ is *SCC-wise non-redundant*.

Let S be an SCC and let

$$\mathcal{R}(S) = \{r \in \mathcal{R} \mid \{head(r)\} \cup body(r) \subseteq S\}$$
$$\mathcal{R}_h(S) = \{r \in \mathcal{R} \mid head(r) \in S\}$$

Definition 47. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework. We define the non-circular ABA D° as follows. For each SCC S and each rule $r = s \leftarrow p_1, \dots, p_n$ in $\mathcal{R}_h(S)$ we consider $k = k(S) = |S|$ copies $r_1, \dots, r_k$ where for each $1 \le j \le k$:*

- *$head(r_j) = s^j$ for $j < k$ and $head(r_k) = s$,*
- *if $p_i \notin S$, then $p_i \in body(r_j)$,*
- *if $p_i \in S$, then $p_i^{j-1} \in body(r_j)$.*

For notational convenience we let i) $p^j = p$ whenever $p \notin S$ and ii) $s^k = s$; hence we have

- *$head(r_j) = s^j$,*
- *$body(r_j) = \{p^j \mid p \in body(r)\}$*

Then we let $D^\circ = (\mathcal{L}^\circ, \mathcal{R}^\circ, \mathcal{A}, ^{-\circ})$ where

$$\mathcal{L}^\circ = \bigcup_{S \in SCCs(D)} \bigcup_{j=1}^{k} \{s^j \mid s \in \mathcal{L}\},$$

$$\mathcal{R}^\circ = \bigcup_{S \in SCCs(D)} \bigcup_{j=1}^{k} \{r^j \mid r \in \mathcal{R}\}$$

$$\overline{p^i}^\circ = \overline{p}^i$$

Lemma 48. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-})$ be an ABA framework and $k \ge 1$ an integer. If there is an SCC-wise non-redundant argument t with $leaves(t) = A$ and $cl(t) = s$ in D, then there is also an argument t' in D° satisfying the same two conditions. Vice versa, if t is an argument in D° with $leaves(t) = A$ and $cl(t) = s$, then there is also an argument t' in D satisfying the same two conditions.*

Proof. ($\Rightarrow$) Take s and suppose $s \in S$ for some SCC. By induction, we assume that claim holds for each parent SCC of S. Suppose A^* is the union of all the assumptions required to construct the sub-trees outside (i.e. before) S. Let $|S| = k$.

Let us call the longest sub-path $P(S)$ in S the derivation depth of $A \vdash s$. (the SCC is fixed for the rest of the proof). For the given integer k we show the claim for each argument $A \vdash s$ with derivation depth $l \le k$; thereby we proceed per induction over l.

More precisely, or inductive hypothesis is as follows:

> If t with $leaves(t) = A$ and $cl(t) = s$ has derivation depth l in D, then t^j with $leaves(t) = A$ and $cl(t) = s^j$ is constructible in D° for each $l \le j \le k$

Let $l = 1$. Hence in D we can derive s via a rule

$$r = s \leftarrow p_1, \dots, p_n$$

s.t. $p_1, \dots, p_n \notin S$. Due to the k copies we construct in D°, we get

$$r^j = s^j \leftarrow p_1, \dots, p_n$$

for each $1 \leq j \leq k$, which proves the claim via inferring the p_i from the previous SCCs (induction).

Now suppose the claim holds for each $l < k$ and consider some argument t with $leaves(t) = A$ and $cl(t) = s$ in D with derivation depth $l + 1 \leq k$. Consider the top-most rule r which is used to derive s. This rule is of the form

$$s \leftarrow p_1, \ldots, p_n.$$

By definition, there are sub-arguments $t_1, \ldots, t_n$ with $leaves(t_i) = A_i$ and $cl(t_i) = p_i$ and again by definition their derivation depth at most l. Note that $A = A_1 \cup \ldots \cup A_t \cup A^*$ by construction of tree-based arguments.

By the inductive hypothesis, we have arguments t_i^j with $leaves(t_i^j) = A_i$ and $cl(t_i^j) = p_i^j$ for each $1 \leq i \leq t$ and each $l \leq j \leq k$. By construction of D° and due to the rule $s \leftarrow p_1, \ldots, p_t$, there is for each $l \leq j \leq k$ a rule with head s^{j+1} and $\{p_1^j, \ldots p_t^j\}$ as body. Using the sub-arguments t^j we found above, we can apply this rule in order to obtain the desired argument t^{j+1} with $leaves(t^{j+1}) = A$ and $cl(t) = s^{j+1}$.

($\Leftarrow$) Again this direction is much easier: In a tree-based argument t in D° replace each rule applied rule

$$s^{j+1} \leftarrow p_1^j, \ldots p_n^j$$

with the original one

$$s \leftarrow p_1, \ldots p_n$$

and suitably adjust the atoms, i.e., replace each s^{j+1} with s and p_i^j with p_i. This way, a valid tree-based argument in D is obtained. □

Corollary 49. *Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-)$ be an ABA. Then D° is non-circular and satisfies $\sigma(D) = \sigma(D^\circ)$ for each $\sigma \in \{adm, com, prf, grd, stb\}$.*

Proof. Non-circularity is by construction. Due to Corollary 46 D° contains all non-redundant arguments and application of Lemma 29 yields $\sigma(D) = \sigma(D^\circ)$. □