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

Anna Rapberger¹, Markus Ulbricht² and Johannes P. Wallner³

¹TU Wien, Institute of Logic and Computation

²Leipzig University, Department of Computer Science

³Graz University of Technology, Institute of Software Technology

Abstract

A key ingredient of computational argumentation in AI is the generation of arguments in favor or against claims under scrutiny. In this paper we look at the complexity of the argument generation procedure in the prominent structured formalism of assumption-based argumentation (ABA). We show several results connecting expressivity of ABA fragments and number of constructed arguments. First, for several NP-hard fragments of ABA, the number of generated arguments is not bounded polynomially. Even under equivalent rewritings of the given ABA framework there are situations where one cannot avoid an exponential blow-up. We establish a weaker notion of equivalence under which this blow-up can be avoided. As a general tool for analyzing ABA frameworks and resulting arguments and their conflicts, we extend results regarding dependency graphs of ABA frameworks, from which one can infer structural properties on the induced attacks among arguments.

1. Introduction

Computational models of argumentation are a central approach within non-monotonic reasoning [1] with a variety of applications [2] in, e.g., legal or medical reasoning. Key to many approaches to computational argumentation are formalisms in what is called structured argumentation which specify formal argumentative workflows, with assumption-based argumentation (ABA) [3], ASPIC⁺ [4], defeasible logic programming (DeLP) [5], and deductive argumentation [6] among the prominent approaches in the field. Reasoning within these formalisms is oftentimes carried out by instantiating argument structures and conflicts among these arguments from (rule-based) knowledge bases in a principled manner. The resulting arguments and (directed) conflicts are referred to as argumentation frameworks (AFs) [7]. Argumentation semantics define argumentative acceptability on an AF s.t. conclusions can be drawn for the original knowledge base.

In the present paper, we will focus on ABA [8] which is well studied and has applications in, e.g., decision making [9, 10, 11]. 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.

Constructing an AF corresponding to a given knowledge base has several advantages. From a technical point of view, there is an abundance of research concerned with AFs (see [1] for an overview) which can be applied to assess the instantiated AF. 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; much information encoded in the knowledge base is made explicit and clear within the graphical framework.

Let us consider the following situation. Suppose we plan to model the behavior of the propositional CNFformula $\phi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$ via an ABA knowledge base (see Section 2 for a formal introduction to ABA). For this, we identify ϕ with the set $\{C_1, C_2\}$ of clauses $C_1 = \{x_1, \neg x_2\}$ and $C_2 = \{\neg x_1, x_2\}$. A natural representation would make use of assumptions corresponding to the four occurring literals, i.e. A = $\{x_1, x_2, x'_1, x'_2\}$. Then, *rules* model satisfaction of the given clauses; we construct

$$r_1 = C_1 \leftarrow x_1. \qquad r_3 = C_2 \leftarrow x'_1.$$
$$r_2 = C_1 \leftarrow x'_2. \qquad r_4 = C_2 \leftarrow x_2.$$

with the intuitive meaning that e.g. C_1 can be derived if r_1 or r_2 is applicable which in turn is the case if the assumption x_1 or the assumption $\neg x_2$ is set to true, respectively. Lastly, the rule " $r_5 = \phi \leftarrow C_1, C_2$." models that ϕ can be derived if both C_1 and C_2 can. When constructing the argumentation framework corresponding to this ABA knowledge base, we would make the conditions under which ϕ can be derived (i.e. what satisfying assignments to the formula exist) visible; indeed, among others, the following arguments would be obtained:

A_1 :	$ \begin{smallmatrix} \phi \\ / \\ c_1 c_2 \\ & \\ x_1 x'_1 \end{smallmatrix} $	A_3 :	$\begin{smallmatrix}\phi\\/\searrow\\c_1c_2\\\downarrow\\x_2'x_1'\end{smallmatrix}$	A_2 :	$ \begin{smallmatrix} \phi \\ / \\ c_1 c_2 \\ & \\ x_1 x_2 \end{smallmatrix} $	A_4 :	$\begin{smallmatrix}\phi\\/\searrow\\c_1\ c_2\\ \ \ x_2'\ x_2\end{smallmatrix}$	
---------	---	---------	---	---------	--	---------	--	--

NMR 2022: 20th InternationalWorkshop on Non-Monotonic Reasoning, August 07–09, 2022, Haifa, Israel 0 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Minimution 40 International (CC BV 40). CEUR Workshop Proceedings (CEUR-WS.org)

We can now directly read off that e.g. $\{x_1, x_2\}$ (see A_3) constitutes a satisfying assignment to ϕ since both C_1 and C_2 (and thus ϕ) can be derived. We can also see that $\{x_1, x_1'\}$ would in principle infer ϕ as well, but this set of literals does not correspond to a well-defined (partial) assignment; we can not set x_1 to true and false simultaneously. We thus see that constructing the arguments helps visualizing the information encoded in the knowledge base and makes certain relations explicit. Indeed, simply inspecting all arguments deriving ϕ suffices to decide whether there is a satisfying assignment to our formula.

However, since checking satisfiability of a given CNFformula is the prototypical NP-complete problem, we expect that this procedure must come with computational cost elsewhere. Indeed, many structured argumentation formalisms (including ABA) suffer from the drawback that the knowledge base gives rise to exponentially many (or even an infinite amount of) arguments [12, 13, 14]. So, in a nutshell, the instantiated AF makes information within the knowledge base explicit, but requires many arguments to do so. This gives rise to the following question:

> Is the argumentation framework induced by an ABA knowledge base large in size, but reasoning is easy?

Although this idea is appealing in principle, the answer turns out to be negative in general: Even if we reduce our attention to the class of AFs induced by ABA knowledge bases, reasoning is still intractable in general. Nonetheless, we expect an underlying trade-off between size and complexity which is worth to be investigated. As a high level observation we expect an ideal ABA knowledge base to

- induce a large AF, but with reasoning being simple, or
- a concise AF, in which reasoning is potentially hard.

If the former is the case, then the instantiation procedure can help displaying relationships encoded within the knowledge base, as it was the case for our CNF formula ϕ from above. If on the other hand the latter case occurs, then the constructed AF is efficient and easy to capture visibly. In this paper, we will investigate this trade-off and examine the relationship between size and complexity of the induced AF. Most notably, we will show that each ABA framework can be transferred in a way that either reasoning becomes easy (Theorem 32) or the AF is of polynomial size (Proposition 28), but reaching both goals simultaneously is not possible in general (Theorem 32).

In doing so, our goal is also contributing to a general formal understanding under which conditions a knowledge base induces large AFs and which techniques might help avoiding this blow-up. For carrying out reasoning on ABA via using AFs, both the size of the resulting AF and complexity of the resulting AF can pose barriers to (automated) argumentative reasoning. On the one hand size and complexity of AFs are direct challenges for solvers [15, 16]. On the other hand, a large number of arguments in an AF can be a barrier to methods supporting argumentative explanations on the AF, due to the sheer number of arguments. Therefore, investigating formal foundations which provide guidance how to encode information, depending on the intended behavior of the induced AF, is a worthwhile endeavor.

The main contributions of the paper can be summarized as follows.

- We first deal with the case of infinite instantiated AFs and show that one can restrict attention to finite cores.
- We relate bounds on the given ABA instance to (i) size bounds and computability of the constructed AF and (ii) complexity of reasoning.
- We show that each ABA framework can be rewritten in a way that reasoning in the induced AF is tractable, while credulous acceptance of a target conclusion is preserved.
- We present, for a fairly large class of ABA frameworks, a general transformation procedure to obtain a translated ABA framework which is equivalent under projection and has a polynomial-sized AF. We present an impossibility result suggesting that the notion of equivalence cannot be strengthened to full equivalence.
- We extend the notion of dependency graphs on ABA frameworks (see, e.g., Craven and Toni, 2016) to a general tool for investigating ABA frameworks able to check structural properties inducing milder complexity, such as acyclic or odd-cycle free AFs.

2. Assumption-based Argumentation

We recall preliminaries for assumption-based argumentation (ABA) [8, 3] and argumentation frameworks (AFs) [7].

The first ingredient of ABA is that of 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 $s \leftarrow a_1, \ldots, a_n$ with $a_i \in \mathcal{L}$. A shorthand for the head of a rule r is defined by $head(r) = \{s\}$, and for the (possibly empty) body via $body(r) = \{a_1, \ldots, a_n\}$. An ABA framework contains a deductive system and specifies which atoms are assumptions and what are contraries of assumptions.

Definition 1. An ABA framework 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).

In this work, we focus on ABA frameworks which are (i) flat, i.e., for each rule $r \in \mathcal{R}$ it holds that $head(r) \notin \mathcal{A}$ (no assumptions can be derived), (ii) \mathcal{L}, \mathcal{R} , and body(r)for each $r \in R$ is finite, and (iii) each rule in \mathcal{R} is stated explicitly (given as input).

Semantics of ABA frameworks can be defined on subsets of the assumptions or via translation to arguments and attacks. Below, we recall both notions.

Arguments in an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ are based on proof trees (derivations). Due to our focus on computational aspects, we will in later sections consider a different representation of arguments, hence we refer to arguments based directly on proof trees as "tree-based arguments". Formally, a tree-based argument, denoted by $A \vdash_{\mathcal{R}} s$, with $A \subseteq \mathcal{A}$ and $s \in \mathcal{L}$ based on D is defined as a finite labeled rooted tree s.t. the root is labeled with s, each leaf is labeled by an assumption $a \in \mathcal{A}$ or a dedicated symbol $\top \notin \mathcal{L}$ s.t. the set of all labels of leaves is A, and 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. Moreover, if $a \in \mathcal{A}$, then $\{a\} \vdash_{\mathcal{R}} a$ is also a tree-based argument (special case without using any derivation rules). In brief, a tree-based argument represents a derivation using rules in \mathcal{R} to derive s starting from assumptions in \mathcal{A} . We say that s is the claim of the tree-based argument. We remark that there can be multiple tree-based arguments with the same set of assumptions and claim.

Given an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{})$, derivability for a set of assumptions $A \subseteq \mathcal{A}$ is defined via $Th_D(A) = \{s \mid \text{there is a tree-based argument } A' \vdash_{\mathcal{R}} s, A' \subseteq A\}$. That is, $Th_D(A)$ contains all atoms that can be derived (via tree-based arguments) using assumptions in A. We omit subscripts D and \mathcal{R} if clear from the context.

We now recall conflicts, admissible sets, and subsequently the corresponding definitions on tree-based arguments.

Definition 2. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework, 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 $b \in B$.

In words, an assumption set A attacks assumption set B if it is possible to derive from A the contrary of some assumption in B.

Definition 3. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework. An assumption set $A \subseteq \mathcal{A}$ is conflict-free in D iff

A does not attack itself. Set A defends assumption set $B \subseteq A$ in D iff for all $C \subseteq A$ that attack B it holds that A attacks C.

In this paper, we focus on the central semantical concept of *admissibility*.

Definition 4. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework. A set of assumptions $A \subseteq \mathcal{A}$ is admissible iff A is conflict-free and A defends itself.

We move on to tree-based arguments.

Definition 5. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework, and $(A \vdash s)$ and $(B \vdash t)$ two tree-based arguments based on D. Tree-based argument $(A \vdash s)$ attacks $(B \vdash t)$ if there is an assumption $b \in B$ s.t. $\overline{b} = s$.

Collecting all tree-based arguments and attacks based on an ABA results in an argumentation framework corresponding to the given ABA.

Definition 6. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework. The pair (\mathbb{A}, \mathbb{R}) is called the argumentation framework (AF) corresponding to D if \mathbb{A} is the set of all treebased arguments based on D, and \mathbb{R} is the set of all attacks based on D.

Below we recall the analogous notions of conflict-freeness, defense and admissibility in AFs.

Definition 7. Let D be an ABA framework, $F = (\mathbb{A}, \mathbb{R})$ the corresponding AF, and $H \subseteq \mathbb{A}$ a set of tree-based arguments of F. The set H is conflict-free (in F) if there are no arguments $(A \vdash s), (B \vdash t) \in H$ s.t. $(A \vdash$ s) attacks $(B \vdash t)$. A conflict-free set H defends an argument $(A \vdash s)$ (in F) if for each argument $(B \vdash t) \in$ \mathbb{A} that attacks $(A \vdash s)$ it holds that there is an argument $(C \vdash u) \in H$ s.t. $(C \vdash u)$ attacks $(B \vdash t)$. Moreover, His admissible (in F) if it is conflict-free and defends itself.

Claims of tree-based arguments are collected via $cl(H) = \{s \mid (A \vdash s) \in H\}$ for a set of treebased arguments H, and assumptions via $asms(H) = \bigcup_{(A \vdash s) \in H} A$.

To clearly distinguish semantics, we say that $A \subseteq A$ is an admissible assumption set (or an *adm*-assumption set) and that a set of tree-based arguments E is an admissible extension (or *adm*-extension for short). We refer to all *adm*-assumption sets of D via adm(D), and to all *adm*-extensions of an AF F by adm(F). There is a direct correspondence between semantics via assumption sets and sets of tree-based arguments (see, e.g., Čyras et al., 2018).

Proposition 8. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ })$ be an ABA framework and $F = (\mathbb{A}, \mathbb{R})$ the corresponding AF.

- If $E \in adm(D)$ then $\{(A \vdash s) \mid A \subseteq E, (A \vdash s) \text{ is a tree-based argument in } D\} \in adm(F)$, and
- if $H \in adm(F)$ then $asms(H) \in adm(D)$.

An important reasoning task on ABA and AFs is credulous reasoning under admissibility. An atom *s* is credulously accepted under admissibility in an ABA *D* iff there is an *adm*-assumption-set *E* s.t. $s \in Th(E)$. For a given AF (\mathbb{A}, \mathbb{R}) and $\alpha \in \mathbb{A}$, it holds that α is credulously accepted under admissibility in *F* iff there is an *adm*extension *H* containing α . 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 *adm*-extension containing some tree-based argument α with claim *s*.

Complexity results for reasoning in ABA and AFs were established (see, e.g., 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, credulous acceptance under admissibility is *NP*-complete.

Example 9. We formalize the introductory example. The ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ is given by

$$\mathcal{L} = \{c_1, c_2, \phi\} \cup \mathcal{A},$$
$$\mathcal{A} = \{x_1, x'_1, x_2, x'_2\} \text{ with } \overline{x_i} = x'_i, \overline{x'_i} = x_i,$$

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 $A \subseteq A$ 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 $\phi \in Th(\{x_1, x_2\})$.

3. Infinite AFs, Cores, and Representation

In general it can be the case that an AF F corresponding to a given ABA D is not finite. We first recall some basic properties of such (possibly) infinite corresponding AFs.

We say that an AF $F = (\mathbb{A}, \mathbb{R})$ is infinite if \mathbb{A} is infinite. An AF F is finitary [7] if it holds that each argument $\alpha \in \mathbb{A}$ is attacked by a finite number of arguments (but the overall number of arguments may still be infinite).

Example 10. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ 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 (countably) many treebased arguments based on D (via chaining rules arbitrary many times), and argument $\{a\} \vdash a$ is attacked by all tree-based arguments concluding y (of which there are infinitely many).

This leads to the simple observation stated next. That there are countably many tree-based arguments can be seen since one can write (for a given ABA) each argument as a string over a restricted alphabet.

Observation 11. *Given an ABA framework, the corresponding AF can be (countably) infinite and non-finitary.*

Nevertheless, as one can see intuitively in the example, AFs corresponding to an ABA framework can be "cut down" to a finite core by removing "duplicates" of arguments. This observation is sometimes assumed to be folklore in the research community and stated for other forms of structured argumentation [12].

Let us formalize how to obtain such a duplicate-free core. Tree-based arguments in an ABA framework are defined as proof trees, with each argument $(A \vdash s)$ based on a set of assumptions and a claim. While rules are driving derivability, they are not important when evaluating arguments: conflicts between arguments are solely specified via assumptions A and claim s. A natural way to represent arguments is thus by using only A and s. From now on, we mean by arguments pairs (A, s) but insist that there is a corresponding proof tree $(A \vdash s)$ in the given ABA framework. We call the resulting set of arguments the *core* of an ABA.

Definition 12. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \neg)$ be an ABA framework. Let $\mathbb{A} = \{(A, s) \mid$ there is a tree-based argument $(A \vdash s)$ in $D\}$. An argument (A, s) attacks an argument (B, t) (in D) if $\exists b \in B \ s.t. \ \bar{b} = s$, with \mathbb{R} being the set of all such attacks. 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: $cl(H) = \{s \mid (A,s) \in H\}$ and $asms(H) = \bigcup_{(A,s)\in H} A$. Given an ABA *D* its corresponding AF *F* and core *F'*, in addition to the core being finite it follows directly that

- for each tree-based argument (A ⊢ s) there is an argument (A, s) and vice versa, and thus
- *F* has an admissible set of tree-based arguments H with cl(H) = S and asms(H) = A iff F' has an admissible set of arguments H', cl(H') = S and 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 construct the corresponding AF (the core) and perform the

standard reasoning tasks. To some extent surprising perhaps, we can find a complexity-theoretic result supporting the intuition that the core is a more efficient 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 13. It is NP-hard to decide whether there is a proof tree from a given set of assumptions to a given claim.

Proof. Let $\phi = c_1 \land \dots \land c_m$ be a Boolean formula in conjunctive normal form over vocabulary $X = \{x_1, \dots, x_n\}$ with $C = \{c_1, \dots, c_m\}$ the set of clauses. We construct an ABA framework D with $\mathcal{A} = C$, atoms C together with literals over X and $\{d_{x_1}, \dots, d_{x_n}\}$, and the following rules (note that " $\neg x$ " is a symbol in ABA and has no meaning attached to the negation sign):

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

Contraries are assigned to literals not appearing in any rules (for the task of argument construction, contraries are not relevant). It can be shown that (C, f) is an argument of the resulting ABA framework iff ϕ is satisfiable.

We want to emphasize that this result does not imply that it is harder to compute the core compared to a corresponding AF, since one can directly extract the core from the corresponding AF. Rather, Proposition 13 formalizes that skipping computation of the proof trees in order to construct the core is a hard task in general.

From now on, we restrict our attention to cores, and assume that, for a given ABA, we operate exclusively on a core, unless explicitly mentioned otherwise.

4. Bounds on Assumption-based Frameworks

In this section we study the impact of bounding certain parts of the input ABA on the complexity of reasoning and size of cores (and corresponding AFs in cases). We consider bounds on derivation-depth and bodies of rules.

When investigating size of cores, we make use of the following definition. Given an ABA D the size of D(|D|) is the length of a (direct) string representation of D. Let \mathcal{D} be a set of ABA frameworks. We say that the cores of \mathcal{D} are polynomially bounded if there is a polynomial p s.t. $p(|D|) \ge |\mathbb{A}|$ for all $D \in \mathcal{D}$ with $F = (\mathbb{A}, \mathbb{R})$ the core of D. We will see below that the cores of the set of all ABA frameworks are not polynomially bounded.

4.1. Bounds and Complexity of Reasoning

First, we consider bounds on (i) the depth of chaining rules and (ii) the size of bodies of rules. We show that restricting derivation-depth or rule-size does not yield milder complexity regarding credulous reasoning under admissibility.

Formally, an ABA D is bounded by k-derivation-depth if each proof tree of D has height at most k (i.e., the longest path from assumptions to claim is at most k). A rule of the form $s \leftarrow b_1, \ldots, b_n$ is bounded by k if $n \le k$; an ABA D is rule-size bounded by k if each rule in D is bounded by k.

In order to show that bounding derivation-depth is intractable, we reduce the Boolean Satisfiability Problem via the following reduction:

Reduction 14. 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 $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ 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 $\neg x \in c_i$.

The resulting ABA framework is bounded by 2derivation-depth as each proof tree has height at most 2. We observe that the presented reduction formalizes our introductory example:

Example 15. Given the CNF-formula $\phi = (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2)$ from the introduction. Following Reduction 14, we obtain an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \neg)$ which contains the assumptions $\mathcal{A} = \{x_1, x_2, x'_1, x'_2\}$ and the rules $\varphi \leftarrow c_1, c_2$ and

- $c_1 \leftarrow x_1, c_1 \leftarrow x_2'$ (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 resulting framework indeed coincides with the introductory example (cf. Example 9).

It can be shown that the special atom ϕ is credulously accepted under admissible semantics in the ABA framework iff the formula ϕ is satisfiable.

Observe that tree-based arguments in the corresponding AF in Reduction 14 have derivation-depth of at most 2. We obtain that credulous reasoning is *NP*-complete



Figure 1: AF instantiation of the ABA framework from Example 15 for the formula $\phi = (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2)$ (cf. Reduction 14).

even when restricting ABA frameworks to be bounded by k-derivation-depth for some constant $k \ge 2$. Moreover, in general Reduction 14 yields cores which are not polynomially bounded by the given ABA framework: for a formula φ with m clauses and k variables per clause, we construct up to k^m many arguments with conclusion φ . Figure 1 depicts the core (equivalent to the corresponding AF) of the constructed ABA framework from Example 15. Here, we have 2^2 many arguments with conclusion φ .

A slight adaption of the reduction shows that credulous reasoning under admissibility remains *NP*-complete, when bounding instead the body-size of the rules by 2.

Theorem 16. Credulous reasoning under admissibility remains NP-hard even if restricted to derivation-depth bounded or rule-size bounded ABA frameworks.

A closer inspection of the ABAs constructed from Reduction 14 points to another class of ABA frameworks with notable properties. In this reduction, we construct ABA frameworks with a symmetric contrary function, i.e., $\bar{a} = b$ iff $\bar{b} = a$ for all $a, b \in A$. We call these ABA frameworks symmetric. First, observe that credulous reasoning in this framework is *NP*-hard (following directly from Theorem 16).

Corollary 17. Credulous reasoning under admissibility is NP-hard for symmetric ABA frameworks.

On the other hand, we observe that the computational hardness in symmetric ABAs stems entirely from the construction of arguments: indeed, constructing the cores results in AFs with $|\mathcal{A}|/2$ many 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. For an example we refer to Figure 1. 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.

Proposition 18. Credulous reasoning under admissibility is decidable in polynomial time in cores of symmetric ABAs.

4.2. Size of the Constructed AF

From identifying arguments as pairs of assumption sets and claims we directly obtain that the number of arguments in cores is bounded by $2^{|\mathcal{A}|} \cdot |\mathcal{L} \setminus \mathcal{A}| + |\mathcal{A}|$ for each ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$. We establish a bound on the number of tree-based arguments that considers derivation-depth and rule-size and show that bounding both derivation-depth and rule-size yields ABAs with polynomially bounded cores.

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

Theorem 19. For each *m*-derivation-depth and *k*-rulesize bounded ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ with $|\{r \in \mathcal{R} \mid head(r) = s\}| \leq l \text{ for all } s \in \mathcal{L}, \text{ there are at}$ most $l^p \cdot |\mathcal{L} \setminus \mathcal{A}| + |\mathcal{A}|$ many tree-based arguments with $p = \sum_{i=0}^{m-1} k^i.$

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 $p = \sum_{i=0}^{m-1} k^i$ and $n = |\mathcal{L} \setminus \mathcal{A}|$. Here, we do not require that the leaves of the trees are labelled as assumptions. Observe that the set of all tree-based arguments 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 $l \cdot x^k$ 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 k. We express this correspondence via the function $f(x) = l \cdot x^k$. The total number of trees with root s constructible from D after m steps is thus given by $f^m(1)$. We show that $f^m(1) = l^{\sum_{i=0}^{m-1} k^i}$ via induction over m.

For m = 1, we have f(1) = l.

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

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

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

Intuitively, exponentiality of the number of tree-based arguments stems from p, and thus from derivation-depth and rule-size. Moreover, it is clear that the result extends to the number of arguments in the cores of derivation-depth and rule-size bounded ABA frameworks. Bounding both parameters thus yields polynomially bounded cores.

Corollary 20. *The cores of the set of ABA frameworks which are both derivation-depth and rule-size bounded by some constant k are polynomially bounded.*

Bounding derivation-depth or rule-size individually, however, does not yield cores that are polynomially bounded.

Proposition 21. The cores of ABA frameworks which are derivation-depth bounded by some constant k are not polynomially bounded. Likewise, the cores of rule-size bounded ABA frameworks are not polynomially bounded.

4.3. Computation of the Core

As the reader might have noticed, having polynomially bounded cores does not (directly) imply the existence of a polynomial-time algorithm that can actually obtain the core in question. We show that for ABA frameworks which are rule-size bounded an algorithm can be obtained that enumerates arguments in polynomial-time regarding size of input and number of arguments. This can be achieved by a direct algorithm: for each rule r with $body(r) = \{s_1, \ldots, s_n\}$ and already computed (tree-based) arguments $\{\alpha_1, \ldots, \alpha_m\}$ loop through each n-sized subset X of $\{\alpha_1, \ldots, \alpha_m\}$ and check whether cl(X) = body(r). If so, attach it to a potential new argument concluding head(r). If this argument is fresh, add it to the output. If $n \leq k$ for a constant k, this search is bounded polynomially by the given ABA and size of core.

Theorem 22. The cores of ABA frameworks whose rulesize is bounded by a constant can be constructed in polynomial time w.r.t. the size of the given ABA and core.

Proof. Consider the following brute-force procedure.

- loop through each rule $r \in \mathcal{R}$ as long as new arguments are added.
 - given $body(r) = \{a_1, \ldots, a_t\}$, for each subset x_1, \ldots, x_t of the already constructed arguments s.t. $cl(x_i) = a_i$
 - * if the corresponding argument $A \vdash head(r)$ is not yet constructed, add it to the list;
 - * otherwise break;

By definition, this algorithm constructs the correct arguments in F. The outer loop over each rule can be left once there was one iteration which did not induce any

new argument. Therefore, after at most $|\mathbb{A}| + 1$ iterations we are done. Each iteration visits $|\mathcal{R}|$ rules. Consider one particular rule r. For each body literal, we need to search through at most $|\mathbb{A}|$ arguments. Since $|body(r)| \leq k$, this is in $\mathcal{O}(|\mathbb{A}|^k)$. In order to decide whether or not the induced argument $A \vdash head(r)$ needs to be added, we again search through the at most $|\mathbb{A}|$ already constructed arguments and compare. In summary, we visit rules at most $(|\mathbb{A}|+1) \cdot |\mathcal{R}|$ times and thereby, we can perform the necessary computations in time in $1\mathcal{O}(|\mathbb{A}|^{k+1})$. \Box

The preceding result together with Corollary 20 yields a procedure to obtain a polynomially bounded core in polynomial time for rule-size bounded and derivationdepth bounded ABA frameworks.

Corollary 23. The cores of the set of rule-size and derivation-depth bounded ABA frameworks can be computed in polynomial time in size of the given ABA framework.

We remark that it is currently not known whether the condition of rule-size boundedness in Theorem 22 can be dropped; we conjecture that the condition is required. In any case, Theorem 22 also yields the result that enumerating arguments in the core of a rule-size bounded ABA framework has incremental polynomial time complexity [19], i.e., one can in polynomial time w.r.t. the given ABA and partially enumerated arguments find a fresh argument or conclude that all arguments were enumerated.

5. Transformations and Complexity Trade-offs

We saw that ABA frameworks may yield a core not polynomially bounded, but with polynomial-time reasoning (under the credulous view and admissibility), e.g., when looking at Reduction 14. In this section we look at possibility results and an impossibility result of *transforming* a given ABA to another ABA framework whose core is polynomially bounded or allows for polynomial-time reasoning.

5.1. Tractable Reasoning in Core

Recall that one of our main goals was to formalize the idea that one could construct ABA frameworks in a way that the induced AF might be large, but yields tractable reasoning in return. As we already mentioned, the ABA framework constructed in the motivating example corresponds to applying Reduction 14. Our intuition is that the ABA frameworks constructed according to this reduction indeed yield AFs with tractable reasoning (though in the potential exponential size of the AF). This intuition can

be confirmed as follows: First, observe hat Reduction 14 yields symmetric ABA frameworks (by definition). Thus by Proposition 18 stating that reasoning in symmetric AFs is indeed tractable, we obtain the desired outcome.

However, this only states that a certain class of ABA frameworks possesses this property. We would like to go one step further and transform an arbitrary given ABA framework s.t. the corresponding AF behaves this way. It turns out that this can be achieved indeed, at least if we restrict our attention to some target conclusion, say $x \in \mathcal{L}$. The underlying idea is surprisingly simple: Suppose we want to transform the ABA framework D. Since the SAT problem is NP-complete, we can in polynomial time construct a formula ϕ which is satisfiable iff x is credulously accepted in D. Now, our Reduction 14 translates ϕ into an ABA framework with the behavior we wish to obtain. All these steps can be performed in polynomial time. This yields the following main theorem.

Theorem 24. For each ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ and $x \in \mathcal{L}$ one can construct an ABA framework D' in polynomial time s.t.

- under admissibility, x is credulously accepted in D iff x is credulously accepted in D',
- reasoning in the corresponding AF F is tractable in the size of F.

We want to mention however that this procedure as given above is rather theoretical. We believe that finding a constructive proof for this result is an exciting future work direction.

5.2. Obtaining a Polynomial Core

Now we turn our attention to the opposite direction: We present a general polynomial-time procedure which under mild conditions transforms a given ABA framework D s.t. (i) the translated framework is equivalent under projection to D and (ii) the core of the translated ABA framework is polynomially bounded. That is, this time we do not try to obtain an AF with tractable reasoning, but we wish to prevent the exponential blow-up which might be caused from the instantiation procedure.

We take the definition of circular tree-based arguments (proof trees) from Craven and Toni (2016).

Definition 25. 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 ABA framework is circular if there is a circular tree-based argument for this framework.

We remark that ABA frameworks obtained from Reduction 14 are non-circular and all atoms are derivable (independently of whether the underlying formula is satisfiable). The transformation is defined as follows. Intuitively, for each conclusion s derivable from a given ABA framework D, we introduce fresh assumptions s_d ('s is derivable') and s_{nd} ('s is not derivable') that simulate derivations. The resulting ABA framework D' contains only rules with assumptions in the body, each of which gives rise to exactly one tree-based argument. Since no rule in D'contains non-assumptions, each proof tree is of height 1.

Definition 26. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \neg)$ be a non-circular ABA framework such that each $s \in \mathcal{L}$ is in $Th_D(\mathcal{A})$. We define $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}', \neg')$ 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 -'.

Let $\mathcal{A}' = \mathcal{A} \cup \{s_d, s_{nd} \mid s \in \mathcal{L} \setminus \mathcal{A}\}$ and $\mathcal{L}' = \mathcal{L} \cup \mathcal{A}'$. Contraries in D' are defined as for D, except for the new assumptions as above. For each rule $r \in \mathcal{R}$, let r'being r except that if body(r) contains a non-assumption s, replace s by s_d . Finally, set $\mathcal{R}' = \{r' \mid r \in \mathcal{R}\}$.

Let us consider an example.

Example 27. Consider an ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ with assumptions $\mathcal{A} = \{a, b\}$, rules \mathcal{R} :

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

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 26, we obtain the corresponding ABA $D' = (\mathcal{L}', \mathcal{R}', \mathcal{A}', \overset{-}{}')$ with assumptions a, b, and additional assumptions p_d , p_{nd} , q_d , q_{nd} , s_d , s_{nd} ; and rules \mathcal{R}' :

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

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, as we show next: admissible assumption sets and derivations are preserved when projecting to \mathcal{A} of the original ABA framework.

Proposition 28. 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



Figure 2: Constructed ABA framework from Example 15 when applying the construction from Definition 26.

• if $E' \in adm(D')$, then $E = E' \cap \mathcal{A} \in adm(D)$ and $Th_{\mathcal{R}}(E) \supseteq Th_{\mathcal{R}'}(E') \cap \mathcal{L}$.

We note that $Th_{\mathcal{R}}(E)$ and $Th_{\mathcal{R}'}(E') \cap \mathcal{L}$ are not equal but in subset-relation in the second bullet since derivations are potentially cut off in the construction of an AFsensitive ABA framework. Consider the ABA frameworks D and D' in Example 27. In D, the assumption a derives p and q while in D', the assumption a derives only q. Thus $Th_{\mathcal{R}}(\{a\}) = \{a, p, q\} \supseteq \{a, q\} = Th_{\mathcal{R}'}(\{a\})(\cap \mathcal{L}).$

From the construction of AF-sensitive frameworks, it follows that the cores are polynomially bounded. Indeed, since each assumption as well as each rule corresponds to precisely one tree-based argument, we obtain that the corresponding AF has at most $|\mathcal{A}'| + |\mathcal{R}'|$ many arguments in the core.

Proposition 29. Let \mathcal{D} be the set of all non-circular ABA frameworks with all atoms derivable, and \mathcal{D}' the set of AF-sensitive ABA frameworks from \mathcal{D} . It holds that the cores of \mathcal{D}' are polynomially bounded.

Example 30. Let us consider again our ABA framework from Example 15. Applying the above construction, we obtain an ABA D' with additional assumptions $(c_1)_{d}$, $(c_1)_{nd}$, $(c_2)_{d}$, $(c_2)_{nd}$; moreover, we replace rule $\varphi \leftarrow$ c_1, c_2 with the rule $\varphi \leftarrow (c_1)_d$, $(c_2)_d$. The resulting AF is given in Figure 2.

We can now gather our results to infer our desired theorem. Due to Proposition 29 the AF-sensitive ABA framework induces an AF with polynomial size and Proposition 28 ensures that applying this constructions preserves admissible reasoning under projection. Combining these insights yields the following central result.

Theorem 31. There is some polynomial p such that each non-circular ABA framework $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \neg)$ with $\mathcal{L} \subseteq Th_D(\mathcal{A})$ can be transformed in polynomial time into an ABA framework D' where

- if $E \in adm(D)$, then there is an $E' \in adm(D')$ with $E = E' \cap \mathcal{A}$ and $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E') \cap \mathcal{L}$,
- if $E' \in adm(D')$, then $E' \cap \mathcal{A} \in adm(D)$ and $Th_{\mathcal{R}}(E) \supseteq Th_{\mathcal{R}'}(E') \cap \mathcal{L}$, and
- the size of the AF corresponding to D' is bounded by p(|D'|).

5.3. Trade-Off: Size and Complexity

In Section 5.1 the central idea was to start off with Reduction 14 modeling a CNF-formula. We could try to do the same here and then apply Theorem 31 in order to avoid the exponential blow-up in size of the AF. However, if we could proceed like this with one-to-one correspondence of the semantics, we would end up solving an NP-hard problem in polynomial time. Therefore, the condition of equivalence under projection in Theorem 31 seems necessary. We present an impossibility result stating that one cannot transform a given ABA framework to an equivalent one with polynomially bounded cores and bounded rule-size, at least under standard complexity-theoretic assumptions.

Theorem 32. Assuming $P \neq NP$, there is no constant k and polynomial-time algorithm which translates a given ABA framework D to another ABA framework D' s.t.

- adm(D) = adm(D'),
- $Th_{\mathcal{R}}(E) = Th_{\mathcal{R}'}(E)$ for $E \in adm(D)$,
- D' has rule-size bounded by k, and
- the cores of the set of ABA frameworks D' translated by the algorithm are bounded polynomially.

Proof. Suppose that such an algorithm T exists. Let $\phi = c_1 \land \cdots \land c_m$ be a Boolean formula in CNF over clauses $C = \{c_1, \ldots, c_m\}$ and Boolean variables $X = \{x_1, \ldots, x_n\}$. Let D be the ABA framework obtained from Reduction 14, and let D' = T(D) be the outcome of the algorithm. Note that D' can be constructed in time polynomial w.r.t. size of ϕ . By Reduction 14 and second item, it holds that ϕ is satisfiable iff there is an admissible assumption set A with $\phi \in Th_{\mathcal{R}'}(A)$.

Each admissible assumption-set B in D satisfies $B \subseteq X \cup \{x' \mid x \in X\}$, moreover, B does not contain both x and x' for any $x \in X$ Also note that each conflict-free assumption-set is admissible since Reduction 14 yields a symmetric ABA framework.

By the first item, D' has the same admissible assumption sets as D'. It holds that there is an argument (B, ϕ) in D' and B does not contain both x and x' for any $x \in X$ iff ϕ is credulously accepted under admissible semantics in D' iff ϕ is credulously accepted under admissible semantics in D iff ϕ is satisfiable. Thus, if one finds an argument (B, ϕ) for D' without "complementary literals" (i.e., without any $x \in X$ s.t. both x and x' are in B) we

conclude that ϕ is satisfiable, and if ϕ is unsatisfiable no such argument in D' exists.

By Theorem 22 and due to the third item, one can enumerate all arguments of D' in polynomial time. Overall, we can search the space of all arguments in D' in time polynomial to size of ϕ , implying that we can decide satisfiability of ϕ in polynomial time, a contradiction to $P \neq NP$.

6. Dependency Graphs on ABA

We look at dependencies induced by the rules of a given ABA framework, inspired by dependency graphs in logic programming, and related but different to the dependency graph notion existing for ABA [17].

Definition 33. The dependency graph $G_D = (V, E, l)$ for a given ABA $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ is an edge-labelled graph with

- $V = \mathcal{L}$ is the set of vertices,
- $edge \ e = (s,t) \in E$ iff i) there is some rule $r \in \mathcal{R}$ with $s \in body(r)$ and $head(r) = \{t\}$, in this case l(e) = +; or ii) $t \in \mathcal{A}$ and $\overline{t} = s$, in this case, l(e) = -.

In brief, vertices represent atoms, and positive edges connect body elements and heads of rules, while negative edges correspond to contraries.

Example 34. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \bar{})$ be an ABA framework with $\mathcal{A} = \{a, b, c, d, e\}$, distinct contraries $\overline{x} = \overline{x}$ for each $x \in \mathcal{A}$ and the following rules:

$$\overline{d} \leftarrow a \qquad \overline{b} \leftarrow a \qquad \overline{c} \leftarrow b \\ \overline{a} \leftarrow c \qquad \overline{e} \leftarrow d \qquad \overline{a} \leftarrow e$$

Then the dependency graph G_D is given as follows.

A path (cycle) in G_D is defined as usual; the length of a path (cycle) is the number of edges labeled "—". The relation between paths in G_D and the core F of D is very close as formalized in the following lemma.

Lemma 35. Let D be an ABA framework with core $F = (\mathbb{A}, \mathbb{R})$ and dependency graph G_D . If $(\alpha, \beta) \in \mathbb{R}$, then there is a path of length one from $cl(\alpha)$ to $cl(\beta)$ in G_D .

The observation can be extended to the following result.

Proposition 36. Let D be an ABA framework with core F and dependency graph G_D . i) If G_D is acyclic, then so is F. ii) If G_D is odd-cycle free, then so is F.

Proof. Suppose there is a cycle in F. Then there is also a cycle x_1, \ldots, x_n in F where $cl(x_i) \neq cl(x_j)$ for $i \neq j$ which can be seen as follows. If $cl(x_i) = cl(x_j)$, then (supposing i < j) there is an attack from x_i to x_{j+1} and we can simply remove the sub-sequence consisting of $x_{i+1}, \ldots x_j$. This procedure can be iterated until we obtain the sub-sequence with the two required properties.

We now prove the two statements. i) By iteratively applying Lemma 35 we find a cycle in G_D . ii) By utilizing $cl(x_i) \neq cl(x_j)$ for $i \neq j$, Lemma 35 even guarantees that the length is preserved which yields an odd cycle in G_D if x_1, \ldots, x_n is odd.

The above proposition does not hold for even cycles. As a counter-example consider the even-cycle free dependency graph of Example 34. In contrast, the core F possesses an even cycle. Vice versa, given a cycle in the dependency graph, we can construct a cycle of the same length in F.

Proposition 37. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework such that each $s \in \mathcal{L}$ is in $Th(\mathcal{A})$, F be the core and G_D the dependency graph. If p_1, \ldots, p_n is a cycle in G_D , then there is a cycle of the same length in F.

Proof. For $2 \le i \le n$ let p_i be an assumption. By construction of G_D , $p_{i-1} = \overline{p_i}$. Since D is trim, there is an argument x_1 with conclusion $cl(x_1) = p_{i-1}$. Now let j be minimal s.t. i < j and p_j is an assumption. Again since D is trim, there is also an argument x_2 with $cl(x_2) = p_{j-1} = \overline{p_j}$. Observe that x_2 is attacked by arguments with conclusion p_{i-1} . Thus, we have an attack from x_1 to , x_2 in F. In G_D , the length of the path p_{i-1}, \ldots, p_{j-1} equals one since by choice of j we have $\{p_{i-1}, \ldots, p_{j-1}\} \cap \mathcal{A} = \{p_i\}$. If we continue analogously we find a cycle in F having the same length as our cycle p_1, \ldots, p_n .

The dependency graph can be utilized to obtain approximations for several bounds we considered throughout this paper.

Proposition 38. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework and $G_D = (V, E, l)$ the dependency graph. i) For the rule size k of D we have $k \leq \max_{e \in V} |\{e' \in V \mid (e', e) \in E\}|$. ii) If G_D is acyclic, then the derivation-depth of D is bounded by the size of the longest path (counting "+" labels) in G_D .

Finally, we get a tighter approximation for the size of the core than our previously established term $2^{|\mathcal{A}|} \cdot |\mathcal{L} \setminus \mathcal{A}|$.

Proposition 39. Let $D = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$ be an ABA framework, $F = (\mathbb{A}, \mathbb{R})$ be the core and G_D the dependency graph. For each $s \in \mathcal{L}$ let $p(s) = \{a \in \mathcal{A} \mid a \text{ has a path in } G_D \text{ to } s\}$. Then $|\mathbb{A}| \leq |\mathcal{A}| + \sum_{s \in \mathcal{L} \setminus \mathcal{A}} 2^{|p(s)|}$.

7. Discussion

The computation of a corresponding AF, and different forms of representing arguments together with optimizations for structured argumentation have been considered before, e.g., for ABA [17, 20, 13], and for other forms of structured argumentation [12, 21]. Moreover, complexity of ABA was investigated in several directions [22, 23, 24, 25], potentially exponential AFs arising from structured argumentation and their issues was discussed, e.g., by Strass et al. (2019), and infinite arguments for ABA were investigated [27]. 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 focused on arguments as pair structures (assumptions and claim).

Considering variants for representing arguments is an appealing avenue for future work, but it seems that different representations arrive at other computational barriers. For instance, Lehtonen et al. (2017) showed #P-hardness under subtractive reductions for counting the number of argument structures for a representation incorporating a form of minimality on the assumption sets. When requiring a core to contain only arguments with subset-minimal assumption sets one can show hardness for argument construction, e.g., it is *NP*-hard to decide whether there is a subset-minimal argument not yet constructed.

Our results on trade-offs regarding size and complexity of reasoning suggest that it might be worthwhile to pre-process ABA frameworks in a suitable way before utilizing AF solvers; also, one may model problems in ABA such that the corresponding AF is either small in size or the reasoning in the AF is tractable, depending on the intention.

The most interesting future work directions we identify are as follows. First, the results which make use of the actual semantics are only phrased for admissible sets yet. In particular generalizing Theorem 31 to the other classical Dung semantics would contribute to our research. Moreover, Theorem 24 is not constructive and one needs to fix a given atom in advance. Finding an analogous construction which preserves the semantics, similar in spirit to Theorem 31, would be a great generalization. Furthermore, it would be interesting to investigate to which extent a pre-processing as suggested by Theorem 31 can help boosting the performance of ABA solvers. As a final remark we want to mention that many other structured formalisms are similar in their spirit, i.e. induce a potentially infinite AF which makes information encoded within the knowledge base explicit. One could therefore examine whether our results translate to other formalisms as well.

Acknowledgments

We thank the anonymous reviewers of this paper for their helpful comments. This research was funded in whole, or in part, by the Austrian Science Fund (FWF) W1255-N23, by the Vienna Science and Technology Fund (WWTF) through project ICT19-065, and by the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig.

References

- P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, College Publications, 2018.
- [2] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, S. Villata, Towards artificial argumentation, AI Magazine 38 (2017) 25–36.
- [3] A. Bondarenko, P. M. Dung, R. A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artif. Intell. 93 (1997) 63–101.
- [4] S. Modgil, H. Prakken, A general account of argumentation with preferences, Artif. Intell. 195 (2013) 361–397.
- [5] A. J. García, G. R. Simari, Defeasible logic programming: An argumentative approach, Theory Pract. Log. Program. 4 (2004) 95–138.
- [6] P. Besnard, A. Hunter, Elements of Argumentation, MIT Press, 2008.
- [7] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (1995) 321–358.
- [8] K. Cyras, X. Fan, C. Schulz, F. Toni, Assumptionbased argumentation: Disputes, explanations, preferences, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, College Publications, 2018, pp. 365–408.
- [9] R. Craven, F. Toni, C. Cadar, A. Hadad, M. Williams, Efficient argumentation for medical decision-making, in: Proc. KR, AAAI Press, 2012, pp. 598–602.
- [10] K. Čyras, T. Oliveira, A. Karamlou, F. Toni, Assumption-based argumentation with preferences and goals for patient-centric reasoning with interacting clinical guidelines, Argument Comput. 12 (2021) 149–189.
- [11] X. Fan, F. Toni, A. Mocanu, M. Williams, Dialogical two-agent decision making with assumptionbased argumentation, in: Proc. AAMAS, IFAA-MAS/ACM, 2014, pp. 533–540.
- [12] L. Amgoud, P. Besnard, S. Vesic, Equivalence in

logic-based argumentation, J. Appl. Non Class. Logics 24 (2014) 181–208.

- interface, Int. J. Approx. Reason. 112 (2019) 55-84.
- [13] T. Lehtonen, J. P. Wallner, M. Järvisalo, From structured to abstract argumentation: Assumption-based acceptance via AF reasoning, in: Proc. ECSQARU, volume 10369 of *LNCS*, Springer, 2017, pp. 57–68.
- [14] B. Yun, N. Oren, M. Croitoru, Efficient construction of structured argumentation systems, in: Proc. COMMA, volume 326 of *FAIA*, IOS Press, 2020, pp. 411–418.
- [15] M. Thimm, S. Villata, The first international competition on computational models of argumentation: Results and analysis, Artif. Intell. 252 (2017) 267– 294.
- [16] S. A. Gaggl, T. Linsbichler, M. Maratea, S. Woltran, Design and results of the second international competition on computational models of argumentation, Artif. Intell. 279 (2020).
- [17] R. Craven, F. Toni, Argument graphs and assumption-based argumentation, Artif. Intell. 233 (2016) 1–59.
- [18] W. Dvořák, P. E. Dunne, Computational problems in formal argumentation and their complexity, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, College Publications, 2018.
- [19] D. S. Johnson, C. H. Papadimitriou, M. Yannakakis, On generating all maximal independent sets, Inf. Process. Lett. 27 (1988) 119–123.
- [20] Z. Bao, K. Čyras, F. Toni, ABAplus: Attack reversal in abstract and structured argumentation with preferences, in: Proc. PRIMA, volume 10621 of *LNCS*, Springer, 2017, pp. 420–437.
- [21] B. Yun, S. Vesic, M. Croitoru, Toward a more efficient generation of structured argumentation graphs, in: Proc. COMMA, volume 305 of *FAIA*, IOS Press, 2018, pp. 205–212.
- [22] Y. Dimopoulos, B. Nebel, F. Toni, On the computational complexity of assumption-based argumentation for default reasoning, Artif. Intell. 141 (2002) 57–78.
- [23] K. Čyras, Q. Heinrich, F. Toni, Computational complexity of flat and generic assumption-based argumentation, with and without probabilities, Artif. Intell. 293 (2021) 103449.
- [24] T. Lehtonen, J. P. Wallner, M. Järvisalo, Declarative algorithms and complexity results for assumptionbased argumentation, J. Artif. Intell. Res. 71 (2021) 265–318.
- [25] A. Karamlou, K. Čyras, F. Toni, Complexity results and algorithms for bipolar argumentation, in: Proc. AAMAS, IFAAMAS, 2019, pp. 1713–1721.
- [26] H. Strass, A. Wyner, M. Diller, *EMIL*: Extracting meaning from inconsistent language: Towards argumentation using a controlled natural language

[27] P. M. Thang, P. M. Dung, J. Pooksook, Infinite arguments and semantics of dialectical proof procedures, Argument Comput. (2021). In-press.