Decomposition-Guided Reductions for Argumentation and Treewidth*

Johannes Fichte[1], **Markus Hecher**[2,3], **Yasir Mahmood**[4], **Arne Meier**[4]
[1]UC Berkeley, Berkeley, USA
[2]TU Wien, Vienna, Austria
[3]University of Potsdam, Germany
[4] Leibniz Universität Hannover, Germany
johannes.fichte@berkeley.edu, hecher@dbai.tuwien.ac.at, {mahmood, meier}@thi.uni-hannover.de

Abstract

Argumentation is a widely applied framework for modeling and evaluating arguments and its reasoning with various applications. Popular frameworks are abstract argumentation (Dung's framework) or logic-based argumentation (Besnard-Hunter's framework). Their computational complexity has been studied quite in-depth. Incorporating treewidth into the complexity analysis is particularly interesting, as solvers oftentimes employ SAT-based solvers, which can solve instances of low treewidth fast. In this paper, we address whether one can design reductions from argumentation problems to SAT-problems while linearly preserving the treewidth, which results in decomposition-guided (DG) reductions. It turns out that the linear treewidth overhead caused by our DG reductions, cannot be significantly improved under reasonable assumptions. Finally, we consider logic-based argumentation and establish new upper bounds using DG reductions and lower bounds.

1 Introduction

Argumentation is a widely applied framework for modeling and evaluating arguments and its reasoning with various applications. Many different directions of argumentation theory have been successfully perused in the area of AI [Amgoud and Prade, 2009; Maher, 2016; Rago *et al.*, 2018]. Popular frameworks are *abstract argumentation* (Dung's framework) [Dung, 1995; Rahwan, 2007] or *logic-based argumentation* [Besnard and Hunter, 2008].

In abstract argumentation, one describes arguments by notions that state acceptability with respect to an abstract framework, such as stable or admissible. Such arguments are then called extensions of a framework. In the logic-based method, one aims for inclusion-minimal consistent sets Φ of formulas (the *support*) that entail a *claim* α, which is encoded by a Boolean formula. If such a pair exists, then one calls (Φ, α) an *argument*. In this context, we consider three central decision problems. The first, ARG, asks, given a set Δ of formulas, the so-called *knowledge-base (KB)*, and a formula α, whether there exists a subset $\Phi \subseteq \Delta$ such that (Φ, α) is an argument in Δ. The two further problems of interest are $\mathrm{ARG\text{-}Check}$, which asks whether a given set is a support for a given claim, and $\mathrm{ARG\text{-}Rel}$, for which—besides given KB and claim—a formula has to be contained in the support.

Example 1. (A1) Support: *We have enough money (x_{em}) and there is no travel restriction, so we can travel (x_{tr}). Claim: We can travel to Montreal (x_{tM}).* (A2) Support: *Corona cases are increasing (x_{C}) and governments are imposing travel restrictions. Claim: We cannot travel to anywhere anymore. Formalizing these yields:* $A_1 : \Phi_1 = \{x_{\mathsf{em}}, x_{\mathsf{tr}}, (x_{\mathsf{em}} \wedge x_{\mathsf{tr}}) \rightarrow x_{\mathsf{tM}}\}, \alpha_1 = \{x_{\mathsf{tM}}\}$, $A_2 : \Phi_2 = \{x_{\mathsf{C}}, x_{\mathsf{C}} \rightarrow \neg x_{\mathsf{tr}}\}, \alpha_2 = \{\neg x_{\mathsf{tr}}\}$. *Each argument supports its claim, whereas, together they are conflicting, as A_2 attacks A_1.*

The *computational complexity* of abstract argumentation has been studied quite in-depth for different problems and fragments of existence [Dunne and Bench-Capon, 2002; Dvořák and Woltran, 2010; Dvořák, 2012], (projected) counting [Baroni *et al.*, 2010; Fichte *et al.*, 2019], and enumeration [Kröll *et al.*, 2017] with results mostly on the first or second level of the polynomial hierarchy. Similarly for logic-based argumentation, ARG was shown to be Σ_2^p-complete [Parsons *et al.*, 2003]. More in-depth works consider the dichotomy between classes of tractability and intractability [Creignou *et al.*, 2011; Creignou *et al.*, 2014]. Also, more fine-grained analyzes when incorporating additional structure have been established [Dvořák *et al.*, 2012; Fichte *et al.*, 2019; Lampis *et al.*, 2018], for example, *treewidth*, which is defined on graph representations of the input. Treewidth k of an instance describes the hardness for evaluating the instance (bucket elimination) [Dechter, 1999; Cygan *et al.*, 2015], when designing an algorithm that avoids backtracking and brute-forces only $f(k)$ times. Treewidth is particularly interesting for analyzing the complexity, as abstract argumentation solvers oftentimes employ solvers based on SAT and extensions [Brochenin *et al.*, 2018; Charwat *et al.*, 2015; Alviano, 2018], which can solve instances of low treewidth fast [Atserias *et al.*, 2011; Bacchus *et al.*, 2003]. However, employing low treewidth only works in practical settings if the reduction to (extensions of) SAT is not already very expensive in the treewidth; more precisely, if there are reductions from problems of abstract argumentation to SAT or 2-QBF that linearly preserve the treewidth and can be

*The work is supported by Austrian Science Fund (FWF), Grants Y698 and P32830, the Vienna Science and Technology Fund, Grant WWTF ICT19-065, and the DFG (ME4279/1-2) under 247444366.

computed reasonably fast. Surprisingly, this is unknown for abstract argumentation. For logic-based argumentation even bounded treewidth results are missing.

Contributions. In more detail, we address these questions:

1. We present decomposition-guided reductions (DG) for abstract argumentation problems when parameterized by treewidth. Our DG reductions are guided by a tree decomposition and allow to compile argumentation problems into SAT or 2-QBF with low (tree)width overhead.
2. We confirm that such reductions cannot be significantly improved under the exponential time hypothesis (ETH).
3. Furthermore, we consider the setting of logic-based argumentation. We establish new upper bounds using DG reductions and lower bounds by reducing from 2-QBF (ARG and ARG-Rel) or SAT (ARG-Check).

Related Works. Upper bounds by reductions to QBF for problems in abstract argumentation and treewidth under admissible and preferred semantics have been considered [Lampis *et al.*, 2018]. Dynamic programming algorithms and lower bounds have been established for various semantics [Fichte *et al.*, 2019]. We go beyond and present a systematic approach for the full set of standard semantics that linearly preserves the treewidth. Hecher [2020] recently established reductions that employ decompositions for answer-set programming. Various works considered QBF and ETH lower bounds for treewidth [Chen, 2004; Lampis and Mitsou, 2017]. Solvers that explicitly exploit treewidth proved useful in various applications, e.g., model counting [Hecher *et al.*, 2020] and QBF [Charwat and Woltran, 2019].

2 Preliminaries

We assume familiarity with computational complexity [Pippenger, 1997], graph theory [Bondy and Murty, 2008], and Boolean logic [Biere *et al.*, 2009].

Quantified Boolean Formulas. Let ℓ be a positive integer, which we call *(quantifier) rank* later, and $\top$ and $\bot$ be the constant always evaluating to 1 and 0, respectively. For a Boolean formula F, we abbreviate by $\mathsf{var}(F)$ the variables occurring in F and $F(X_1, \ldots, X_l)$ to indicate that $X_1, \ldots, X_l \subseteq \mathsf{var}(F)$. A *quantified Boolean formula* ϕ (in prenex normal form), *qBf* for short, is an expression of the form $\phi = Q_1X_1.Q_2X_2.\cdots Q_\ell X_\ell.F(X_1, \ldots, X_\ell)$, where for $1 \leq i \leq \ell$, we have $Q_i \in \{\forall, \exists\}$ and $Q_i \neq Q_{i+1}$, the X_i are disjoint, non-empty sets of Boolean variables, and F is a Boolean formula. We let $\mathsf{matrix}(\phi) := F$ and we say that ϕ is *closed* if $\mathsf{var}(\mathsf{matrix}(F)) = \bigcup_{i\in\ell} X_i$. We evaluate ϕ by $\exists x.\phi \equiv \phi[x \mapsto 1] \vee \phi[x \mapsto 0]$ and $\forall x.\phi \equiv \phi[x \mapsto 1] \wedge \phi[x \mapsto 0]$ for a variable x. W.l.o.g. we assume that $\mathsf{matrix}(\phi) = \psi_{\text{CNF}} \wedge \psi_{\text{DNF}}$, where ψ_{CNF} is in CNF (disjunction of conjunctions of literals) and ψ_{DNF} is in DNF (conjunction of disjunctions of literals). Then, depending on Q_ℓ, either ψ_{CNF} or ψ_{CNF} is optional, more precisely, ψ_{CNF} might be $\top$, if $Q_\ell = \forall$, and ψ_{DNF} is allowed to be $\top$, otherwise. The problem ℓ-QBF asks, given a closed qBf $\phi = \exists X_1.\phi'$ of rank ℓ, whether $\phi \equiv 1$ holds. The problem $\#\ell$-QBF asks, given a closed qBf $\exists X_1.\phi$ of rank ℓ, to count assignments α to X_1 such that $\phi[\alpha] \equiv 1$. For brevity we might omit ℓ.

Tree Decompositions and Treewidth. For a rooted (directed) tree $T = (N, A)$ with *root* $\mathrm{root}(T)$ and a node $t \in N$, we let $\mathsf{children}(t)$ be the set of all nodes t', which have an edge $(t, t') \in A$. Let $G = (V, E)$ be a graph. A *tree decomposition (TD)* of a graph G is a pair $\mathcal{T} = (T, \chi)$, where T is a rooted tree, and χ is a mapping that assigns to each node t of T a set $\chi(t) \subseteq V$, called a *bag*, such that:

1. $V = \bigcup_{t \text{ of } T} \chi(t)$ and $E \subseteq \bigcup_{t \text{ of } T}\{\{u, v\} \mid u, v \in \chi(t)\}$
2. for each s lying on any r-t-path: $\chi(r) \cap \chi(t) \subseteq \chi(s)$.

Then, define $\mathsf{width}(\mathcal{T}) := \max_{t \text{ of } T} |\chi(t)|-1$. The *treewidth* $\mathsf{tw}(G)$ of G is the minimum $\mathsf{width}(\mathcal{T})$ over all tree decompositions $\mathcal{T}$ of G. Observe that for every vertex $v \in V$, there is a unique node t^* with $v \in \chi(t^*)$ such that either $t^* = \mathrm{root}(T)$ or there is a node t of T with $\mathsf{children}(t)=\{t^*\}$ and $v \notin \chi(t)$. We refer to the node t^* by $\mathsf{last}(v)$. For arbitrary but fixed $w \geq 1$, it is feasible in linear time to decide if a graph has treewidth at most w and, if so, to compute a TD of width w [Cygan *et al.*, 2015]. In this work, we assume only TDs (T, χ), where for every node t of T, we have that $|\mathsf{children}(t)| \leq 2$. Such a TD can be obtained from any TD in linear time without increasing the width.

Treewidth and qBfs. For a given qBf ϕ with $\mathsf{matrix}(\phi) = \psi_{\text{CNF}} \wedge \psi_{\text{DNF}}$, we define the *primal graph* $\mathcal{G}_\phi = \mathcal{G}_{\mathsf{matrix}(\phi)}$, whose vertices are $\mathsf{var}(\mathsf{matrix}(\phi))$. Two vertices of $\mathcal{G}_\phi$ are adjoined by an edge, whenever the corresponding variables share a clause or term of ψ_{CNF} or ψ_{DNF}, respectively.

Let $\mathsf{tower}(i, p)$ be $\mathsf{tower}(i-1, 2^p)$ if $i > 0$ and p otherwise. Further, we assume that $\mathsf{poly}(n)$ is any polynomial for given positive integer n. The following result is known for QBF.

Proposition 2 (Chen, 2004)**.** *For any arbitrary qBf ϕ of quantifier rank $\ell > 0$, the problem ℓ-QBF can be solved in time* $\mathsf{tower}(\ell, \mathcal{O}(\mathsf{tw}(\mathcal{G}_\varphi))) \cdot \mathsf{poly}(|\mathsf{var}(\phi)|)$.

Assuming the *exponential time hypothesis (ETH)* [Impagliazzo *et al.*, 2001], one cannot significantly improve this runtime in the worst case. Intuitively, the ETH implies that neither SAT=1-QBF nor #SAT=#1-QBF can be decided in time better than $2^{o(|\mathsf{var}(\varphi)|)}$ for an arbitrary formula φ.

Proposition 3 (Fichte *et al.*, 2020)**.** *Under ETH, for any arbitrary qBf φ of quantifier rank $\ell > 0$, problem ℓ-QBF cannot be solved in time* $\mathsf{tower}(\ell, o(\mathsf{tw}(\mathcal{G}_\varphi))) \cdot \mathsf{poly}(|\mathsf{var}(\varphi)|)$.

Abstract Argumentation. We use Dung's argumentation framework [Dung, 1995] and consider only non-empty and finite sets of arguments A. An *(argumentation) framework (AF)* is a directed graph $F = (A, R)$ where A is a set of arguments and $R \subseteq A \times A$, a pair of arguments representing direct attacks of arguments. An argument $s \in S$, is called *defended by S in F* if for every $(s', s) \in R$, there exists $s'' \in S$ such that $(s'', s') \in R$. The family $\mathrm{def}_F(S)$ is defined by $\mathrm{def}_F(S) := \{s \mid s \in A, s \text{ is defended by } S \text{ in } F\}$. In abstract argumentation, one strives for computing so-called *extensions*, which are subsets $S \subseteq A$ of the arguments that have certain properties. The set S of arguments is called *conflict-free in S* if $(S \times S) \cap R = \emptyset$; S is *admissible in F* if (1) S is *conflict-free in F*, and (2) every $s \in S$ is *defended by S in F*. Let $S_R^+ := S \cup \{ a \mid (b, a) \in R, b \in S \}$ and S be admissible. Then, S is a) *complete in F* if $\mathrm{def}_F(S) = S$; b) *preferred in* F, if no $S' \supset S$ exists that is *admissible in* F; c) *semi-stable in* F if no admissible set $S' \subseteq A$ in F with $S^+_R \subsetneq (S')^+_R$ exists; and d) *stable in* F if every $s \in A \setminus S$ is *attacked* by some $s' \in S$. A conflict-free set S is *stage in* F if there is no conflict-free set $S' \subseteq A$ in F with $S^+_R \subsetneq (S')^+_R$. For a semantics $\mathcal{S} \in \{\mathsf{adm}, \mathsf{comp}, \mathsf{pref}, \mathsf{semiSt}, \mathsf{stab}, \mathsf{stag}\}$, we write $\mathcal{S}(F)$ for the set of *all extensions* of semantics $\mathcal{S}$ in F. Given an AF $F = (A, R)$. Then, problem $\mathcal{S}$ asks if $\mathcal{S}(F) \neq \emptyset$ and $\#\mathcal{S}$ asks to compute $|\mathcal{S}(F)|$. The problems $c_\mathcal{S}$ and $s_\mathcal{S}$ question for $c \in A$ and $s \in A$, whether c is in some $S \in \mathcal{S}(F)$ ("*credulously* accepted") and s is in every $S \in \mathcal{S}(F)$ ("*skeptically* accepted"), respectively. Finally, problem $\#c_\mathcal{S}$ asks for $c \in A$, to compute $|\{S \mid S \in \mathcal{S}(F), c \in S\}|$.

TDs for AFs. Consider for AF $F = (A, R)$ the *primal graph* $\mathcal{G}_F$, where we simply drop the direction of every edge, i.e., $\mathcal{G}_F = (A, R')$ where $R' := \{\{u, v\} \mid (u, v) \in R\}$. For any TD $\mathcal{T} = (T, \chi)$ of $\mathcal{G}_F$ and any node t of T, we let $R_t := R \cap \{(a, b) \mid a, b \in \chi(t)\}$ be the *bag attacks of* t.

Logic-based Argumentation. We study *logic-based argumentation (LA)* [Besnard and Hunter, 2008] using notions of Creignou et al. [2014]. Given sets Φ and Δ of Boolean formulas and a Boolean formula α, the tuple (Φ, α) is an *argument for* α if (1) Φ is consistent, (2) $\Phi \models \alpha$, and (3) Φ is subset-minimal w.r.t. (2). In case of $\Phi \subseteq \Delta$, tuple (Φ, α) is an *argument in* Δ. We call α the *claim*, Φ the *support* of the argument, and Δ the *knowledge-base*. We consider the following problems. The problem ARG (argument existence) asks, given a set of formulas Δ and a formula α, is is there a set $\Phi \subseteq \Delta$ such that (Φ, α) is an argument in Δ? The problem ARG-Check (verification) asks, given a set of formulas Φ and a formula α, is is (Φ, α) an argument? The problem ARG-Rel (relevance) asks, given a set of formulas Δ, and formulas $\psi \in \Delta$ and α, is is there a set $\Phi \subseteq \Delta$ with $\psi \in \Phi$ such that (Φ, α) is an argument in Δ? Note that for deciding ARG and ARG-Rel, Condition (3) above is irrelevant.

In this work, we assume w.l.o.g. that the KB Δ is a set of clauses (CNF) and that α is in DNF. This simplifies presentation, but is not a hard restriction, as problems ARG and ARG-Rel remain Σ^p_2-, whereas ARG-Check remains **DP**-complete [Parsons *et al.*, 2003; Creignou *et al.*, 2011].

TDs for LA. In order to apply treewidth for logic-based argumentation, we let Δ be a set of clauses and α be a Boolean formula in DNF. Then, the *primal graph* $\mathcal{G}_{(\Delta,\alpha)} := \mathcal{G}_{\Delta \wedge \alpha}$, where Δ is viewed as a CNF formula. Further, for any TD $\mathcal{T} = (T, \chi)$ of $\mathcal{G}_{(\Delta,\alpha)}$ and any node t of T, let $\Delta_t := \{\phi_i \mid \phi_i \in \Delta, \mathsf{var}(\phi_i) \subseteq \chi(t)\}$ be the *bag knowledge base of* t.

Example 4. *Consider the argument* (Φ_1, α_1) *from Example 1. Notice that the support* Φ_1 *can be written in CNF as* $\Phi_1 = \{x_{\mathsf{em}}, x_{\mathsf{tr}}, \neg x_{\mathsf{em}} \vee \neg x_{\mathsf{tr}} \vee x_{\mathsf{tM}}\}$ *and* $\alpha_1 = \{x_{\mathsf{tM}}\}$. *Then,* $\mathcal{G}_{(\Phi,\alpha)}$ *has* $\{x_{\mathsf{em}}, x_{\mathsf{tr}}, x_{\mathsf{tM}}\}$ *as the set of vertices and there are edges between all three, because they share a clause.*

3 Decomposition-Guided Reductions for AFs

We briefly discuss a new type of reductions below.

3.1 Decomposition-Guided Reductions to QBF

Inspired by recent related work [Hecher, 2020], we introduce so-called decomposition-guided reductions as follows.

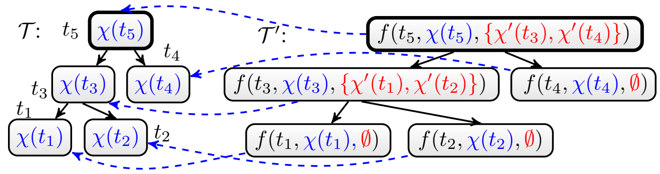

Figure 1: Illustration of a DG reduction $\mathcal{R}$ from problem P to QBF, where we take an instance $\mathcal{I}$ of problem P and a TD $\mathcal{T} = (T, \chi)$ of $\mathcal{G}_\mathcal{I}$. Then, since the DG reduction is constructed for each node t of T, it immediately yields a TD $\mathcal{T}' = (T, \chi')$ of $\mathcal{G}_\varphi$ of the resulting qBf φ. Each bag $\chi'(t)$ of a node t of T functionally depends on t, $\chi(t)$, as well as $\chi'(t')$ of every child node $t' \in \mathsf{children}(t)$.

A *decomposition-guided (DG) reduction* $\mathcal{R}$ is a function that takes both an instance $\mathcal{I}$ of a problem P and a TD $\mathcal{T} = (T, \chi)$ of $\mathcal{G}_\mathcal{I}$, and returns a qBf φ in time $\mathsf{tower}(\ell, o(\mathsf{width}(\mathcal{T}))) \cdot \mathsf{poly}(|\mathsf{var}(\varphi)|)$. The time restriction ensures that $\mathcal{R}$ does not already solve the resulting qBf (cf. Proposition 2). The way a DG reduction is constructed, it has to yield a TD $\mathcal{T}' = (T, \chi')$ of $\mathcal{G}_\varphi$. So, the idea of such a DG reduction is to construct φ from a TD node's point of view. Thereby, for each node t of T, the constructed bag $\chi'(t)$ functionally depends on the original bag $\chi(t)$, but also on the constructed bags $\chi'(t_1), \ldots, \chi'(t_o)$ of its child nodes $\{t_1, \ldots, t_o\} = \mathsf{children}(t)$. This gives rise to a function f that takes a tree decomposition node t, its bag $\chi(t)$ and a set $\chi'(\mathsf{children}(t)) := \{\chi'(t_i) \mid t_i \in \mathsf{children}(t)\}$ of constructed bags for the child nodes of t. Figure 1 illustrates that function f taking a node t, its original bag $\chi(t)$, as well as $\chi'(\mathsf{children}(t))$, to construct each bag $\chi'(t) = f(t, \chi(t), \chi'(\mathsf{children}(t)))$. Then, since $\mathsf{width}(\mathcal{T})$ is bounded by $\mathcal{O}(\max_{t \text{ of } T}(|\chi(t)|))$, also the treewidth of the resulting qBf is at most $\mathcal{O}(\max_{t \text{ of } T}(|f(t, \chi(t), \chi'(\mathsf{children}(t))|))$. Intuitively, DG reductions are *guided* by a TD $\mathcal{T} = (T, \chi)$ and adhere to ideas of dynamic programming along TD $\mathcal{T}$. However, the DG reduction has to ensure that $\mathcal{T}'$ is a TD of $\mathcal{G}_\varphi$.

Next, we present DG reductions for problems originating from abstract argumentation that linearly preserve treewidth. These problems serve as a demonstration of DG reductions. To this end, we assume for the rest of this section that $F = (A, R)$ is an AF and that $\mathcal{T} = (T, \chi)$ is a given TD of $\mathcal{G}_F$.

3.2 Stable Extensions

First, we compute stable extensions via a reduction to SAT, i.e., such that the extensions are represented via a Boolean formula. To this end, we start with the following reduction. We use a variable e_a for every argument $a \in A$, which indicates whether a is in the extension or not. These variables are part of the *extension variables* $E := \{e_a \mid e_a \in A\}$. We use sub-formulas $\mathsf{conf}_R(E)$ and $\mathsf{inOrX}_R(E)$ to ensure conflict-freeness and to determine that every argument is either in the extension or attacked by the extension, respectively. More formally, we let $\mathsf{conf}_R(E) := \bigwedge_{(a,b) \in R}(\neg e_a \vee \neg e_b)$ and $\mathsf{inOrX}_R(E) := \bigwedge_{a \in A}(\bigvee_{(b,a) \in R} e_b \vee e_a)$.

These definitions can be used to encode stab, but also $\#c_{\mathsf{stab}}$: $\exists E.\mathsf{conf}_R(E) \wedge \mathsf{inOrX}_R(E) \wedge e_c$. While $\mathsf{conf}_R(E)$ already preserves the treewidth, $\mathsf{inOrX}_R(E)$ does not. This is witnessed by the observation that formula $\mathsf{inOrX}_R(E)$ could cause dense parts in the primal graph of the formula.

DG Reduction. Consequently, one needs to split $\mathsf{inOrX}_R(E)$ in order to linearly preserve the treewidth. To this end, we design a DG reduction, where we also consider the TD $\mathcal{T}$. We split $\mathsf{inOrX}_R(E)$ with the help of auxiliary variables of the form d_a^t for every node t of T and argument $a \in A$ to indicate whether a is attacked ("defeated") by an argument $b \in \chi(t)$ of the extension. This leads to *defeated variables* $D := \{d_a^t \mid a \in A, t \text{ in } T\}$.

Then, we define a DG reduction $\mathcal{R}_{\#c_{\mathsf{stab}}\rightarrow\#\mathrm{SAT}}(F, \mathcal{T})$ by

$$\mathcal{R}_{\#c_{\mathsf{stab}}\rightarrow\#\mathrm{SAT}}(F, \mathcal{T}) := \exists E, D.\varphi_{\#\mathsf{stab}}(E, D) \wedge e_c,$$

where $\varphi_{\#\mathsf{stab}}(E, D)$ is a CNF consisting of Formulas (1)–(3):

$$d_a^t \leftrightarrow \bigvee_{\substack{t' \in \mathsf{children}(t),\\ a \in \chi(t')}} d_a^{t'} \vee \bigvee_{(b,a) \in R_t} e_b \qquad \text{for every } t \text{ of } T, a \in \chi(t) \quad (1)$$

$$\mathsf{conf}_R(E) \quad (2)$$

$$e_a \vee d_a^{\mathsf{last}(a)} \qquad \text{for every } a \in A. \quad (3)$$

Intuitively, Formulas (1) guide information of defeated arguments along the TD and Formulas (3) ensure that an argument is either in the extension or defeated. Since the reduction is constructed for each node of T, it is easy to see that the DG reduction is correct and preserves the (tree)width linearly.

Theorem 5 (TW-Awareness)**.** *Let $F = (A, R)$ be an AF and $\mathcal{T} = (T, \chi)$ be a TD of F of width k. Then, the DG reduction $\mathcal{R}_{\#c_{\mathsf{stab}}\rightarrow\#\mathrm{SAT}}(F, \mathcal{T})$ constructs a qBf ψ that linearly preserves the width, i.e.,* $\mathsf{tw}(\mathcal{G}_{\mathsf{matrix}(\psi)}) \in \mathcal{O}(k)$.

Proof (Sketch). We construct a TD $\mathcal{T}' = (T, \chi')$ of $\mathcal{G}_\psi$ as follows. For every node t of T, we let $\chi'(t) := \chi(t) \cup \{e_a, d_a^t \mid a \in \chi(t)\} \cup \{d_a^{t'} \mid a \in \chi(t) \cap \chi(t'), t' \in \mathsf{children}(t)\}$. Since $|\mathsf{children}(t)| \leq 2$, we have that $|\chi'(t)| \leq 5 \cdot |\chi(t)|$. ☐

Interestingly, it is not expected that one can significantly improve (decrease) the treewidth in such a DG reduction.

Theorem 6 (TW-LB)**.** *Let $F = (A, R)$ be an AF and $\mathcal{T}$ be a TD of F. Under ETH, the DG reduction $\mathcal{R}_{\#c_{\mathsf{stab}}\rightarrow\#\mathrm{SAT}}(F, \mathcal{T})$ cannot be significantly improved, i.e., there is no reduction $\mathcal{R}'$ from c_{stab} to* SAT *yielding a qBf ψ in time $2^{o(\mathsf{tw}(\mathcal{G}_F))} \cdot \mathsf{poly}(|A|)$ with $\mathsf{tw}(\mathcal{G}_\psi) \in o(\mathsf{tw}(\mathcal{G}_F))$.*

Proof (Sketch). Assume towards a contradiction that there is such a reduction $\mathcal{R}'$. Then, we apply $\mathcal{R}'$ in order to solve c_{stab} for F in time $2^{o(\mathsf{tw}(\mathcal{G}_F))} \cdot \mathsf{poly}(|A|)$, which contradicts ETH [Fichte *et al.*, 2019]. ☐

3.3 Admissible Extensions

Similar to above, for computing admissible extensions, we follow a plain reduction to SAT. As above, we use extension variables E as well as sub-formula $\mathsf{conf}_R(E)$ to determine conflict-freeness. Further, we require $\mathsf{def}_R(E)$ to ensure that attackers of the extension are defeated, respectively. Let $\mathsf{def}_R(E) := \bigwedge_{(b,a) \in R}(\bigvee_{(c,b) \in R} e_c \vee \neg e_a)$.

Then, these definitions can be used to encode adm as follows: $\exists E.(\mathsf{conf}_R(E) \wedge \mathsf{def}_R(E))$. While $\mathsf{conf}_R(E)$ already preserves the treewidth, $\mathsf{def}_R(E)$ does not, which is witnessed by the same argument as for $\mathsf{inOrX}_R(E)$ above.

DG Reduction. Towards a DG reduction, one needs to split $\mathsf{def}_R(E)$ in order to linearly preserve the treewidth. Similar to above, we split $\mathsf{def}_R(E)$ with the help of auxiliary variables, namely the defeated variables D. However, we also need further auxiliary variables of the form n_a for every argument $a \in A$ to indicate whether a never attacks an argument of the extension. These variables are referred to by the *no-attacking variables* $N := \{n_a \mid a \in A\}$.

Then, we define DG reduction $\mathcal{R}_{\mathsf{adm}\rightarrow\mathrm{SAT}}(F, \mathcal{T})$ to SAT

$$\mathcal{R}_{\mathsf{adm}\rightarrow\mathrm{SAT}}(F, \mathcal{T}) := \exists E, D, N.\varphi_{\mathsf{adm}}(E, D, N), \text{ where}$$

$\varphi_{\mathsf{adm}}(E, D, N)$ consists of Formulas (1), (2) and (4), (5):

$$\neg n_a \vee \neg e_b \qquad \text{for every } (a, b) \in R \quad (4)$$

$$e_a \vee n_a \vee d_a^{\mathsf{last}(a)} \qquad \text{for every } a \in A. \quad (5)$$

Thereby, Formulas (4) define n_a and Formulas (5) generalize Formulas (3) towards admissible semantics.

Lifting to $\#c_{\mathsf{adm}}$. In order to bijectively preserve admissible extensions of F, we add to $\varphi_{\mathsf{adm}}(E, D, N)$ the following Formulas (6), (7), which finally results in $\varphi_{\#\mathsf{adm}}(E, D, N)$:

$$\neg n_a \vee \neg e_a \qquad \text{for every } a \in A \quad (6)$$

$$\neg n_a \vee \neg d_a^{\mathsf{last}(a)} \qquad \text{for every } a \in A. \quad (7)$$

So, $\mathcal{R}_{\#c_{\mathsf{adm}}\rightarrow\#\mathrm{SAT}}(F, \mathcal{T}) := \exists E, D, N.\varphi_{\#\mathsf{adm}}(E, D, N) \wedge e_c$. As before, this reduction linearly preserves the (tree)width and it is not expected that the treewidth increase can be significantly reduced. Consequently, we obtain similar results to Theorems 5 and 6 for problems $\#c_{\mathsf{adm}}$ and c_{adm}, which can be solved by enforcing that the argument of concern is in the extension. We can further lift the reduction for problems c_{comp} and $\#c_{\mathsf{comp}}$, as shown in an extended (self-archived) version.

3.4 Preferred Extensions

Reusing the definitions from above, one can reduce pref to 2-QBF, where we use a set $\tilde{E}$ of fresh variables obtained from E s.t. $\tilde{E} := \{\tilde{e}_a \mid e_a \in E\}$, by: $\exists E.\forall \tilde{E}.[\mathsf{conf}_R(E) \wedge \mathsf{def}_R(E) \wedge (\tilde{E} \not\supset E \vee \neg \mathsf{conf}_R(\tilde{E}) \vee \neg \mathsf{def}_R(\tilde{E}))]$, where encoding $\tilde{E} \not\supset E$ accordingly is not difficult.

Towards a DG reduction, we keep reusing $\mathsf{conf}_R(\tilde{E})$ over variables $\tilde{E}$. However, checking equality needs to be guided along the TD. We use *inequality variables* q_a (q^t), indicating that $e_a \leftrightarrow \neg \tilde{e}_a$ (for some $a \in A$ below t), respectively. We define $Q := \{q^t, q_a \mid t \text{ of } T, a \in A\}$. Eventually, use fresh sets $\tilde{D}$, $\tilde{N}$ of variables obtained from D, N, respectively.

Then, we define a DG reduction $\mathcal{R}_{\#\mathsf{pref}\rightarrow\#2\text{-}\mathrm{QBF}}(F, \mathcal{T}) :=$

$$\exists E, D, N.\forall \tilde{E}, \tilde{D}, \tilde{N}, Q.[\varphi_{\#\mathsf{adm}}(E, D, N) \wedge (\varphi_{\tilde{E} \not\supset E}(E, \tilde{E}, Q) \vee \neg \varphi_{\#\mathsf{adm}}(\tilde{E}, \tilde{D}, \tilde{N}))],$$

where $\varphi_{\#\mathsf{adm}}$ is constructed as above and $\varphi_{\tilde{E} \not\supset E}(E, \tilde{E}, Q)$ is in DNF consisting of Formulas (8)–(11):

$$e_a \wedge \neg \tilde{e}_a \qquad \text{for every } a \in A \quad (8)$$

$$\neg(q_a \leftrightarrow \tilde{e}_a \wedge \neg e_a) \qquad \text{for every } a \in A \quad (9)$$

$$\neg(q^t \leftrightarrow \bigvee_{t' \in \mathsf{children}(t)} q^{t'} \vee \bigvee_{a \in \chi(t)} q_a) \qquad \text{for every } t \text{ of } T \quad (10)$$

$$\neg q^{\mathrm{root}(T)}. \quad (11)$$

Formulas (8) allow the extension over E to contain an argument not in $\tilde{E}$, Formulas (9) define inequality and Formulas (10) guide inequality along the TD. Finally, with Formulas (11), equal extensions over E and $\tilde{E}$ are allowed. These formulas can be converted to DNF easily, where, e.g., Formulas (9) result in $(q_a \wedge \neg\tilde{e}_a) \vee (q_a \wedge e_a) \vee (\tilde{e}_a \wedge \neg e_a \wedge \neg q_a)$.

As before, the (tree)width is preserved linearly. While c_{pref} can be already decided via c_{adm}, one can solve s_{pref} with this reduction above by adding clause $\neg e_s$ and inverting the result.

3.5 Semistable and Stage Extensions

The idea from above can be reused in order to design DG reduction $\mathcal{R}_{\#c_{\mathsf{semiSt}} \rightarrow \#2\text{-QBF}}(F, \mathcal{T}) :=$

$$\exists E, D, N. \forall \tilde{E}, \tilde{D}, \tilde{N}, Q.[\varphi_{\#\mathsf{adm}}(E, D, N) \wedge e_c \\ (\varphi_{\tilde{E}_R^+ \not\supseteq E_R^+}(E, \tilde{E}, Q) \vee \neg\varphi_{\#\mathsf{adm}}(\tilde{E}, \tilde{D}, \tilde{N}))],$$

where $\varphi_{\tilde{E}_R^+ \not\supseteq E_R^+}(E, \tilde{E}, Q)$ is in DNF and consists of Formulas (10), (11), and (12)–(18) below. Similar to Formulas (8), Formulas (12) and (13) allow that E_R^+ might contain arguments not in $\tilde{E}_R^+$:

$$e_a \wedge \neg\tilde{e}_a \wedge \neg\tilde{d}_a^{\mathsf{last}(a)} \quad \text{for every } a \in A \tag{12}$$

$$d_a^{\mathsf{last}(a)} \wedge \neg\tilde{e}_a \wedge \neg\tilde{d}_a^{\mathsf{last}(a)} \quad \text{for every } a \in A \tag{13}$$

Similar to Formulas (9), we define q_a for ranges of E and $\tilde{E}$, corresponding to $\neg(q_a \leftrightarrow (\tilde{e}_a \vee \tilde{d}_a^{\mathsf{last}(a)}) \wedge \neg e_a \wedge \neg d_a^{\mathsf{last}(a)})$:

$$q_a \wedge \neg\tilde{e}_a \wedge \neg\tilde{d}_a^{\mathsf{last}(a)} \quad \text{for every } a \in A \tag{14}$$

$$q_a \wedge e_a \quad \text{for every } a \in A \tag{15}$$

$$q_a \wedge d_a^{\mathsf{last}(a)} \quad \text{for every } a \in A \tag{16}$$

$$\tilde{e}_a \wedge \neg e_a \wedge \neg d_a^{\mathsf{last}(a)} \wedge \neg q_a \quad \text{for every } a \in A \tag{17}$$

$$\tilde{d}_a^{\mathsf{last}(a)} \wedge \neg e_a \wedge \neg d_a^{\mathsf{last}(a)} \wedge \neg q_a \quad \text{for every } a \in A \tag{18}$$

DG Reduction for $\#c_{\mathsf{stage}}$. Analogously to above, we immediately obtain a DG reduction from $\#c_{\mathsf{stage}}$ to #2-QBF by

$$\mathcal{R}_{\#c_{\mathsf{stage}} \rightarrow \#2\text{-QBF}}(F, \mathcal{T}) := \exists E. \forall \tilde{E}, Q.[\mathsf{conf}_R(E) \wedge e_c \wedge \\ (\varphi_{\tilde{E}_R^+ \not\supseteq E_R^+}(E, \tilde{E}, Q) \vee \neg\mathsf{conf}_R(\tilde{E}))].$$

Indeed, these reductions are correct and treewidth-aware.

Theorem 7 (Correctness)**.** *Given an AF $F = (A, R)$ and a TD $\mathcal{T} = (T, \chi)$ of $\mathcal{G}_F$. Then, the DG reduction $\mathcal{R}_{\#c_{\mathcal{S}} \rightarrow \#2\text{-QBF}}$ for $\mathcal{S} \in \{\mathsf{semiSt}, \mathsf{stage}\}$ is correct, i.e., $\#c_{\mathcal{S}}$ on F coincides with #2-QBF on $\mathcal{R}_{\#c_{\mathcal{S}} \rightarrow \#2\text{-QBF}}(F, \mathcal{T})$.*

Theorem 8 (TW-Awareness TW-LB)**.** *Given an AF $F = (A, R)$ and a TD $\mathcal{T}=(T, \chi)$ of $\mathcal{G}_F$ of width k. Then, the DG reduction $\mathcal{R}_{\#c_{\mathcal{S}} \rightarrow \#2\text{-QBF}}(F, \mathcal{T})$ for $\mathcal{S} \in \{\mathsf{semiSt}, \mathsf{stage}\}$ constructs qBf ψ with $\mathsf{tw}(\mathcal{G}_{\mathsf{matrix}(\psi)}) \in \mathcal{O}(k)$. Under ETH, there is no reduction from $c_{\mathcal{S}}$ to 2-QBF yielding qBf ψ in time $\mathsf{tower}(2, o(\mathsf{tw}(\mathcal{G}_F))) \cdot \mathsf{poly}(|A|)$ with $\mathsf{tw}(\mathcal{G}_\psi) \in o(\mathsf{tw}(\mathcal{G}_F))$.*

4 A Complexity Study for Logic-Based Arg.

4.1 Argument Existence and Relevance

We reduce an instance (Δ, α) of ARG to an instance of 2-QBF. Let $\Delta = \{C_i, | \ 1 \leq i \leq n\}$ be a collection of clauses and α be a Boolean formula in DNF. We use a variable e_i for each i to encode whether C_i is contained in the support. Consequently, let the *support variables* E be defined by $E := \{e_i \mid 1 \leq i \leq n\}$. Then, let M be the set of variables over $\mathsf{var}(\Delta)$. Moreover, let $N := \mathsf{var}(\Delta \cup \{\alpha\})$ and let $\tilde{N}$ denote the renaming of variables in N. That is $\tilde{N} := \{\tilde{x}_i \mid x_i \in N\}$ and each $\tilde{x}_i$ is a fresh variable. Finally, by $\tilde{C}$, $\tilde{\alpha}$ and $\tilde{\Delta}$ denote C, α and Δ over renamed variables. Now, let $\mathsf{cons}_\Delta(E, M) := \bigwedge_{C_i \in \Delta}(\neg e_i \vee C_i)$ and $\mathsf{ent}_{\Delta,\alpha}(E, \tilde{N}) := \bigvee_{C_i \in \Delta}(e_i \wedge \neg\tilde{C}_i) \vee \tilde{\alpha}$. Then, we construct $\psi'_{\mathrm{ARG}} = \exists E, M. \forall \tilde{N}.(\mathsf{cons}_\Delta(E, M) \wedge \mathsf{ent}_{\Delta,\alpha}(E, \tilde{N}))$.

Intuitively, setting $e_i \in E$ to 1 implies that the corresponding $C_i \in \Delta$ constitutes a support Φ. Then, $\mathsf{ent}_{\Delta,\alpha}(E, \tilde{N})$ achieves that whenever an assignment over variables $\tilde{N}$ is a model of each $\tilde{C}_i$, the assignment also models $\tilde{\alpha}$.

Theorem 9 (Correctness)**.** *Let (Δ, α) be an instance of* ARG. *Then, there is a support $\Phi \subseteq \Delta$ such that (Φ, α) is an argument in Δ if and only if ψ'_{ARG} is true.*

The two subformulas $\mathsf{cons}_\Delta(E, M)$ and $\mathsf{ent}_{\Delta,\alpha}(E, \tilde{N})$ do not preserve the treewidth (one bag may contain variables from multiple clauses causing many e_i). For this reason, we split the formula in order to linearly preserve the treewidth.

DG Reduction. Let $\mathcal{T} = (T, \chi)$ be a TD of $\mathcal{G}_{(\Delta,\alpha)}$. Then, we define a *labeled TD (LTD)* $\mathcal{T}' = (T, \chi, \delta)$ of $\mathcal{T}$, where *labeling* $\delta \colon T \rightarrow \Delta$ is such that $\delta(t) \in \Delta_t$ and $\Delta = \bigcup_{t \text{ of } T}\{\delta(t)\}$. Note that an LTD can be easily obtained from any TD without changing the width by copying nodes accordingly. Next, we assume such an LTD $\mathcal{T}' = (T, \chi, \delta)$ of $\mathcal{T}$. Then, we construct the following formula,

$$\mathcal{R}_{\mathrm{ARG} \rightarrow 2\text{-QBF}}(\mathcal{I}, \mathcal{T}') := \exists E, M. \forall \tilde{N}. \varphi_{\mathrm{ARG}}(E, M, \tilde{N}),$$

where φ_{ARG} is built for every node t of T by $\mathsf{cons}_{\{\delta(t)\}}(E, M) \wedge \mathsf{ent}_{\{\delta(t)\}}(E, \tilde{N})$.

Indeed, both subformulas preserve the treewidth linearly.

Theorem 10 (TW-Awareness)**.** *Let $\mathcal{I} = (\Delta, \alpha)$ be an instance of* ARG, *$\mathcal{T}$ be a TD of $\mathcal{G}_{(\Delta,\alpha)}$ of width k, and $\mathcal{T}'$ be an LTD of $\mathcal{T}$. Then, the DG reduction $\mathcal{R}_{\mathrm{ARG} \rightarrow 2\text{-QBF}}(\mathcal{I}, \mathcal{T}')$ constructs a qBf ψ_{ARG} with $\mathsf{tw}(\mathcal{G}_{\mathsf{matrix}(\psi_{\mathrm{ARG}})}) \in \mathcal{O}(k)$.*

Proof (Sketch). For any given LTD $\mathcal{T}'=(T, \chi, \delta)$ of $\mathcal{T}$, the reduction yields a TD $\mathcal{T}_{2\text{-QBF}}=(T, \chi')$ where the set T remains unchanged. Bag $\chi'(t)$ for each node t is constructed by adding a renamed copy of variables in $\chi(t)$ and the support variable e_i to $\chi(t)$, where e_i is such that $\delta(t) = C_i$. □

This immediately yields the following runtime result.

Theorem 11 (Runtime UB)**.** *Let $\mathcal{I} = (\Delta, \alpha)$ be an instance of* ARG. *Then,* ARG *can be solved in time* $\mathsf{tower}(2, \mathcal{O}(k)) \cdot \mathsf{poly}(|\mathsf{var}(\Delta) \cup \mathsf{var}(\alpha)|)$, *where $k = \mathsf{tw}(\mathcal{G}_{\mathcal{I}})$.*

Proof. We construct a TD $\mathcal{T}$ of $\mathcal{G}_{\mathcal{I}}$ of treewidth at most $5 \cdot k$ in time $2^{\mathcal{O}(k)} \cdot \mathsf{poly}(|\mathsf{var}(\Delta) \cup \mathsf{var}(\alpha)|)$ [Cygan *et al.*, 2015] and LTD $\mathcal{T}'$ of $\mathcal{T}$. Then, we use reduction $\mathcal{R}_{\mathrm{ARG} \rightarrow 2\text{-QBF}}(\mathcal{I}, \mathcal{T}')$, which together with Proposition 2 establishes the result. □

Unluckily, this can probably not be significantly improved.

Theorem 12 (Runtime LB)**.** *Let $\mathcal{I} = (\Delta, \alpha)$ be an instance of* ARG. *Then, under ETH,* ARG *cannot be solved in time* $\mathsf{tower}(2, o(\mathsf{tw}(\mathcal{G}_{\mathcal{I}})) \cdot \mathsf{poly}(|\mathsf{var}(\Delta) \cup \mathsf{var}(\alpha)|)$.

	Abstract Argumentation			Logic-based Argumentation	
	$c_{\mathsf{stab}}, c_{\mathsf{adm}}, c_{\mathsf{comp}}, c_{\mathsf{pref}}$, $\#c_{\mathsf{stab}}, \#c_{\mathsf{adm}}, \#c_{\mathsf{comp}}$	s_{pref} / $\#c_{\mathsf{pref}}$	$c_{\mathsf{semiSt}}, c_{\mathsf{stage}}$, $\#c_{\mathsf{semiSt}}, \#c_{\mathsf{stage}}$	ARG ARG-Rel	ARG-Check
TW-Awareness	$\boldsymbol{O(k)}^*$	$\boldsymbol{O(k)}^*$	$\boldsymbol{O(k)}$	$\boldsymbol{O(k)}$	$\boldsymbol{O(k)}$
TW-LB (ETH)	$\boldsymbol{\Omega(k)}$	$\boldsymbol{\Omega(k)}$ / open	$\boldsymbol{\Omega(k)}$	$\boldsymbol{\Omega(k)}$	$\boldsymbol{\Omega(k)}$
Runtime UB	$2^{\mathcal{O}(k)} \cdot \mathsf{poly}(n)$	$2^{2^{\mathcal{O}(k)}} \cdot \mathsf{poly}(n)$	$2^{2^{\mathcal{O}(k)}} \cdot \mathsf{poly}(n)$	$\boldsymbol{2^{2^{\mathcal{O}(k)}} \cdot \mathsf{poly}(n)}$	$\boldsymbol{2^{\mathcal{O}(k)} \cdot \mathsf{poly}(n)}$
Runtime LB (ETH)	$2^{o(k)} \cdot \mathsf{poly}(n)$	$2^{2^{o(k)}} \cdot \mathsf{poly}(n)$ / open	$2^{2^{o(k)}} \cdot \mathsf{poly}(n)$	$\boldsymbol{2^{2^{o(k)}} \cdot \mathsf{poly}(n)}$	$\boldsymbol{2^{o(k)} \cdot \mathsf{poly}(n)}$

Table 1: Overview of results, where $k = \mathsf{tw}(\mathcal{G}_F)$ and $n = |A|$ for given AF $F = (A, R)$ of abstract arg., and $k = \mathsf{tw}(\mathcal{G}_{(\Delta,\alpha)})$, $n = |\mathsf{var}(\Delta) \cup \mathsf{var}(\alpha)|$ for an instance (Δ, α) of logic-based arg. Bold results form new contributions; for known results see [Lampis *et al.*, 2018; Fichte *et al.*, 2019]. *: While DG reductions preserving widths are new, reductions for linearly preserving treewidth are known for adm and pref [Lampis *et al.*, 2018]. "TW-Awareness" refers to the treewidth increase caused by DG reductions, "TW-LB (ETH)" refers to treewidth lower bounds of DG reductions under ETH, "UB" are runtime upper bounds, and "LB (ETH)" are runtime lower bounds under ETH.

One cannot expect to improve the DG reduction much.

Theorem 13 (TW-LB)**.** *Let $\mathcal{I} = (\Delta, \alpha)$ be an instance of* ARG*. Under ETH, there is no reduction $\mathcal{R}'$ from* ARG *to* 2-QBF *yielding a qBf ψ_{ARG} in time $\mathsf{tower}(2, o(\mathsf{tw}(\mathcal{G}_\mathcal{I}))) \cdot \mathsf{poly}(|\mathsf{var}(\Delta) \cup \mathsf{var}(\alpha)|)$ with $\mathsf{tw}(\mathcal{G}_{\psi_{\text{ARG}}}) \in o(\mathsf{tw}(\mathcal{G}_\mathcal{I}))$.*

Argument Relevance Problem (ARG-Rel**).** Consider the reduction for ARG again. The question of ARG-Rel now reduces to forcing one particular element of E in the solution. W.l.o.g., assume that $\psi = C_1$ with $C_1 \in \Delta$. Then, $(\Delta, \alpha, \psi) \in$ ARG-Rel if and only if there is a support Φ that contains C_1. This can be encoded by $\exists E, M.\forall \tilde{N}.(e_1 \wedge \varphi_{\text{ARG}}(E, M, \tilde{N}))$. The correctness proof, treewidth preservation, as well as upper bounds remain the same as for ARG.

Furthermore, notice that the problem ARG-Rel is as hard as the problem ARG. This is because an instance (Δ, α) of ARG has a support Φ if and only if an instance $(\Delta \cup \{\psi\}, \alpha, \psi)$ of ARG-Rel has one, where $\psi \notin \Delta$ is a satisfiable formula over fresh variables not in $\mathsf{var}(\Delta)$. This implies that lower bounds under ETH transfer to ARG-Rel.

4.2 Argument Verification Problem

It seems challenging to reduce a given instance $\mathcal{I} = (\Phi, \alpha)$, where $|\Phi| = n$, of ARG-Check to *one instance* of SAT as ARG-Check is **DP**-complete. A direct reduction encoding all three sub-questions (Φ is consistent, $\Phi \models \alpha$ and no proper subset of Φ entails α) gives a 2-QBF instance. In the following, we reduce (Φ, α) to a collection of qBfs where each subformula contains one quantifier, but no alternation within each subformula. Then, we argue that the resulting qBfs have (tree)width linear in the width of a given TD $\mathcal{T}$ of $\mathcal{G}_{(\Phi,\alpha)}$.

Let $M := \mathsf{var}(\Phi) \cup \mathsf{var}(\alpha)$. In order to encode three separate conditions, we use $n + 1$ many additional copies of variables of M, which we address with $\tilde{M}$ and $M^{\sim i}$ for $1 \leq i \leq n$. Finally, for a formula ϕ (resp., a clause C_j) over M, we write $\tilde{\phi}$ ($\tilde{C}_j$) and $\phi^{\sim i}$ ($C_j{}^{\sim i}$) for the corresponding formula (clause) over $\tilde{M}$ and $M^{\sim i}$, respectively.

Then, $\psi_{\text{ARG-Check}}$ consists of the $n+2$ qBfs of the following three forms. (i) $\exists M.\Phi$, (ii) $\forall \tilde{M}.(\bigvee_{1 \leq i \leq n} \neg \tilde{C}_i \vee \tilde{\alpha})$ and for $1 \leq i \leq n$, (iii) $\exists M^{\sim i}.(\theta_i{}^{\sim i} \wedge \neg \alpha^{\sim i})$, where $\theta_i = \Phi \backslash \{C_i\}$, that is, the formula obtained from Φ by removing the i-th clause. The matrix of the Formula (ii) is in DNF, which encodes that for each assignment over variables in M, either it does not satisfy some clause $C_i \in \Phi$ or it satisfies α. Formulas (iii) encode that for each $C_i \in \Phi$ there is an assignment over M, such that the formula θ_i does not entail α. Notice that all $n + 2$ formulas are independent of each other, this is because each is constructed over a different set of variables. Further, one can merge Formula (i) with Formulas (iii) into one qBf, which together with (ii) results in two qBfs. By slightly abusing notation, we refer to these two qBfs (conjunction) by $\mathcal{R}_{\text{ARG-Check}\to\text{QBF}}(\mathcal{I}, \mathcal{T}) := \psi_{\text{ARG-Check}}$. Notice that reduction $\mathcal{R}_{\text{ARG-Check}\to\text{QBF}}$ consists of several DG reductions corresponding to Formulas (i), (ii), and (iii).

Theorem 14 (Correctness)**.** *Let (Φ, α) be an* ARG-Check*-instance. Then, (Φ, α) is an argument iff $\psi_{\text{ARG-Check}}$ is true.*

The treewidth is preserved (independent copy variables).

Theorem 15 (TW-Awareness)**.** *Let $\mathcal{I}{=}(\Phi, \alpha)$ be an instance of* ARG-Check *and $\mathcal{T} = (T, \chi)$ be a TD of $\mathcal{G}_\mathcal{I}$ of width k. Then, $\mathcal{R}_{\text{ARG-Check}\to\text{QBF}}(\mathcal{I}, \mathcal{T})$ constructs a qBf $\psi_{\text{ARG-Check}}$ with $\mathsf{tw}(\mathcal{G}_{\mathsf{matrix}(\psi_{\text{ARG-Check}})}) \in \mathcal{O}(k)$.*

This implies that ARG-Check can be solved in time $2^{\mathcal{O}(\mathsf{tw}(\mathcal{G}_\mathcal{I}))} \cdot \mathsf{poly}(|\mathsf{var}(\Phi) \cup \mathsf{var}(\alpha)|)$. Moreover, we obtain the matching LB results, similar to Theorem 12 and and 13.

Theorem 16 (Runtime LB)**.** *Let $\mathcal{I} = (\Phi, \alpha)$ be an instance of* ARG-Check*. Then, under ETH,* ARG-Check *cannot be solved in time $2^{o(\mathsf{tw}(\mathcal{G}_\mathcal{I}))} \cdot \mathsf{poly}(|\mathsf{var}(\Phi) \cup \mathsf{var}(\alpha)|)$.*

5 Conclusions

Our results (summarized in Table 1) provide new theoretical insights and strengthen the applicability of solvers that implicitly solve instances by means of tree decompositions. They might also be helpful for solvers that are indirectly able to solve instances of low treewidth fast. We present new treewidth-aware lower bounds (under ETH) as well as tight upper complexity bounds for logic-based argumentation.

As future work, we plan to study practical implementation of this framework and thereby further verify its robustness. Another investigation regarding the strong ETH might underline the strength of our approach. As the reductions preserve the solutions bijectively, they are applicable in the enumeration complexity setting [Fomin and Kratsch, 2010], however a more rigorous approach might lead to further insights. Could other parameters obey similar types of reductions?

References

[Alviano, 2018] Mario Alviano. The Pyglaf Argumentation Reasoner. In *ICLP'17*.

[Amgoud and Prade, 2009] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *Artif. Intell.*, 173(3-4), 2009.

[Atserias *et al.*, 2011] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *J. Artif. Intell. Res.*, 40, 2011.

[Bacchus *et al.*, 2003] Fahiem Bacchus, Shannon Dalmao, and Toniann Pitassi. Algorithms and complexity results for #SAT and Bayesian inference. In *FOCS'03*.

[Baroni *et al.*, 2010] Pietro Baroni, Paul E. Dunne, and Giuseppe De Giacomo, editors. In *COMMA'10*.

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

[Biere *et al.*, 2009] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *FAIA*. IOS Press, 2009.

[Bondy and Murty, 2008] John A. Bondy and Uppaluri S. R. Murty. *Graph theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, 2008.

[Brochenin *et al.*, 2018] Remia Brochenin, Thomas Linsbichler, Marcoa Maratea, Johannes P. Wallner, and Stefan Woltran. Abstract solvers for Dung's argumentation frameworks. *Argument & Computation*, 9(1), 2018.

[Charwat and Woltran, 2019] Günther Charwat and Stefan Woltran. Expansion-based QBF solving on tree decompositions. *Fundam. Inform.*, 167(1-2), 2019.

[Charwat *et al.*, 2015] Günther Charwat, Wolfgang Dvořák, Sarah A. Gaggl, Johannes P. Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation – a survey. *Artif. Intell.*, 220, 2015.

[Chen, 2004] Hubie Chen. Quantified constraint satisfaction and bounded treewidth. In *ECAI'04*.

[Creignou *et al.*, 2011] Nadia Creignou, Johannes Schmidt, Michael Thomas, and Stefan Woltran. Complexity of logic-based argumentation in Post's framework. *Arg. & Comp.*, 2(2-3), 2011.

[Creignou *et al.*, 2014] Nadia Creignou, Uwe Egly, and Johannes Schmidt. Complexity classifications for logic-based argumentation. *ACM Trans. Comput. Log.*, 15(3), 2014.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Marcin Pilipczuk Dániel Marx, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Dechter, 1999] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, 113(1), 1999.

[Dung, 1995] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2), 1995.

[Dunne and Bench-Capon, 2002] Paul E. Dunne and Trevor J. M. Bench-Capon. Coherence in finite argument systems. *Artif. Intell.*, 141(1/2):187–203, 2002.

[Dvořák, 2012] Wolfgang Dvořák. *Computational aspects of abstract argumentation*. PhD thesis, TU Wien, 2012.

[Dvořák and Woltran, 2010] Wolfgang Dvořák and Stefan Woltran. Complexity of semi-stable and stage semantics in argumentation frameworks. *Inf. Proc. Letters*, 110(11), 2010.

[Dvořák *et al.*, 2012] Wolfgang Dvořák, Reinhard Pichler, and S. Woltran. Towards fixed-parameter tractable algorithms for abstract argumentation. *Artif. Intell.*, 186, 2012.

[Fomin and Kratsch, 2010] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. EATCS Series. Springer, 2010.

[Fichte *et al.*, 2019] Johannes K. Fichte, Markus Hecher, and Arne Meier. Counting complexity for reasoning in abstract argumentation. In *AAAI'19*.

[Fichte *et al.*, 2020] Johannes K. Fichte, Markus Hecher, and Andreas Pfandler. Lower Bounds for QBFs of Bounded Treewidth. In *LICS'20*.

[Hecher *et al.*, 2020] Markus Hecher, Patrick Thier, and Stefan Woltran. Taming high treewidth with abstraction, nested dynamic programming, and database technology. In *SAT'20*.

[Hecher, 2020] Markus Hecher. Treewidth-aware reductions of normal ASP to SAT - is normal ASP harder than SAT after all? In *KR'20*.

[Impagliazzo *et al.*, 2001] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4), 2001.

[Kröll *et al.*, 2017] Markus Kröll, Reinhard Pichler, and Stefan Woltran. On the complexity of enumerating the extensions of abstract argumentation frameworks. In *AAAI'17*.

[Lampis and Mitsou, 2017] Michael Lampis and Valia Mitsou. Treewidth with a quantifier alternation revisited. In *IPEC'17*.

[Lampis *et al.*, 2018] Michael Lampis, Stefan Mengel, and Valia Mitsou. QBF as an alternative to courcelle's theorem. In *SAT'18*.

[Maher, 2016] Michael J. Maher. Resistance to corruption of strategic argumentation. In *AAAI'16*.

[Parsons *et al.*, 2003] Simon Parsons, Michael J. Wooldridge, and Leila Amgoud. Properties and complexity of some formal inter-agent dialogues. *J. Logic Comput.*, 13(3), 2003.

[Pippenger, 1997] Nicholas Pippenger. *Theories of computability*. Cambridge University Press, 1997.

[Rago *et al.*, 2018] Antonio Rago, Oana Cocarascu, and Francesca Toni. Argumentation-based recommendations: Fantastic explanations and how to find them. In *IJCAI'18*.

[Rahwan, 2007] Iyad Rahwan. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15), July 2007.