Structure-Aware Encodings of Argumentation Properties for Clique-width*

Yasir Mahmood†1, Markus Hecher‡2, Johanna Groven§3, Johannes K. Fichte¶3

[1]Data Science Group, Heinz Nixdorf Institute, Paderborn University, Germany
[2]CNRS, UMR 8188, Centre de Recherche en Informatique de Lens (CRIL), University of Artois, France
[3]Department of Computer and Information Science, Linköping University, Sweden

Abstract

Structural measures of graphs, such as treewidth, are central tools in computational complexity resulting in efficient algorithms when exploiting the parameter. It is even known that modern SAT solvers work efficiently on instances of small treewidth. Since these solvers are widely applied, research interests in compact encodings into (Q)SAT for solving and to understand encoding limitations. Even more general is the graph parameter clique-width, which unlike treewidth can be small for dense graphs. Although algorithms are available for clique-width, little is known about encodings.

We initiate the quest to understand encoding capabilities with clique-width by considering *abstract argumentation*, which is a robust framework for reasoning with conflicting arguments. It is based on directed graphs and asks for computationally challenging properties, making it a natural candidate to study computational properties. We design novel reductions from argumentation problems to (Q)SAT. Our reductions linearly preserve the clique-width, resulting in directed decomposition-guided (DDG) reductions. We establish novel results for all argumentation semantics, including counting. Notably, the overhead caused by our DDG reductions cannot be significantly improved under reasonable assumptions.

1 Introduction

Many problems in combinatorics, symbolic AI, and knowledge representation and reasoning are computationally very hard [25, 46, 21]. In the literature, various structural restrictions have been identified under which problems become tractable [37, 18, 16]. In theory, we are interested whether an efficient algorithm exists [11, 7, 29, 13]. In practice, we seek effective practical algorithms [36, 34]. From both perspectives, we want to understand the structural combinatorial core of the problem. Structure preserving reductions to

*Authors are stated in reverse alphabetical order.
†yasir.mahmood@uni-paderborn.de
‡hecher@mit.edu
§johanna.groven@liu.se
¶johannes.fichte@liu.se

other problems support this understanding by its algorithmic and logic-based definability character. Important structural properties of input instances are for example hierarchical graph decompositions, which are quite interesting for algorithmic purposes and for solving problems in polynomial-time in the input size and exponential in a parameter defined on the decomposition, e.g., treewidth [30, 17].

Even more general than treewidth is the graph parameter *clique-width* [14], which can even be small for dense graphs where treewidth is large, measuring the distance from co-graphs [15]. Dynamic programming algorithms for deciding acceptance under preferred semantics are known for clique-width [22], but little is known about structure guided reductions to other problems. This is particularly interesting when reasoning relies on tools such as SAT solving where research increasingly asks for efficient encodings and theoretical limitations [32] under the light that logic-based characterizations are known that allow to solve SAT faster.

Proposition 1 ([29]). *For Boolean formulas of directed incidence clique-width k and size n, counting SAT can be solved in time $2^{\mathcal{O}(k)} \cdot poly(n)$.*

The proposition immediately yields the natural research question: *Can we encode problems into (Q)SAT while preserving the clique-width?* Indeed, such encodings would allow to easily reuse Proposition 1 for solving. In this paper, we address this question for abstract argumentation, a natural knowledge representation framework widely used for reasoning with conflicting arguments [19, 44]. There, relationships between arguments are specified in directed graphs, so-called argumentation frameworks (AFs), and conditions are placed on sets (extensions) of arguments that allow AFs to be evaluated. The computational complexity of argumentation is well-studied [22, 24, 21, 9, 27]. Since AFs are already given as direct graphs and the complexity is often even beyond NP, it makes it a perfect candidate to study clique-width aware encoding.

Our main contributions are as follows:

1. We design reductions from argumentation problems to satisfiability of (quantified) Boolean formulas. Our reductions are *directed decomposition-guided (DDG)*, employ k-expressions, and linearly preserve clique-width.

2. For *all* common *argumentation semantics*, we establish favorable upper bounds (tractability) for *extensions existence*, *argument acceptance*, and *counting*. This also works for the maximization-based (second-level) semantics and demonstrates the flexibility of our approach. For these, we rely on an auxiliary result that establishes how one can convert mixed normal-form QBF matrices into DNF and CNF (for inner-most $\forall$ and $\exists$), respectively.

3. We show that the overhead caused by our DDG reductions cannot be significantly improved under reasonable assumptions providing *structurally optimal reductions*. Indeed, this already holds for skeptical reasoning and then immediately carries over to the counting problem.

Related Works Abstract argumentation is widely studied in knowledge representation and reasoning [19, 44, 1, 43]. The computational complexity depends on the considered semantics and common decision tasks range between polynomial-time and the second level of the polynomial hierarchy. For example, deciding whether a given argument belongs to some extension (credulous acceptance) is **NP**-complete for stable semantics and $\boldsymbol{\Sigma}_2^p$-complete for the semi-stable semantics [20, 23, 21]. Counting complexity in abstract argumentation is well-studied [3, 27] with complexity reaching $\#\cdot\mathbf{P}$ (admissible, complete, stable) and $\# \cdot \mathrm{co}\mathbf{NP}$ (preferred, semi-stable, stage) for extension counting and $\# \cdot \mathbf{NP}$ (admissible, complete, stable)

	$\mathsf{s}_\sigma(\mathsf{c}^\star_\sigma)/\#_\sigma$	
$\sigma \in$	$\{\mathsf{stab}, \mathsf{adm}, \mathsf{comp}\}$	$\{\mathsf{pref}, \mathsf{semiSt}, \mathsf{stage}\}$
CW-Aw.	$\mathcal{O}(k)$	$\mathcal{O}(k)$
CW-LB (ETH)	$\Omega(k)$	$\Omega(k)$
Rt (UB/LB)	$2^{\theta(k)} \cdot poly(n)$	$2^{2^{\theta(k)}} \cdot poly(n)$
Ref. (UB)	Thm 7–13	Thm 15–19
Ref. (LB)	Thm. 21/Corr. 22	Prop. 23

Table 1: Overview of our results, where $k = \mathsf{dcw}(\mathcal{G}^d_i(F))$, and $n = |A|$ for given AF $F = (A, R)$. We use c_σ for credulous acceptance, s_σ for skeptical acceptance, and $\#_\sigma$ is the extension counting problem. "CW-Aw." refers to the clique-width increase caused by DDG reductions. "CW-LB (ETH)" refers to clique-width lower bounds of DDG reductions under ETH, "Rt (UB/LB)" are runtime upper and ETH lower bounds. $\star$: Results for $\mathsf{c}_{\mathsf{stab}}$ also apply to $\mathsf{exist}_{\mathsf{stab}}$ whereas exist_σ is trivial for all other semantics. Finally, $\mathsf{c}_{\mathsf{pref}}$ can be solved faster via $\mathsf{c}_{\mathsf{adm}}$ and therefore shares the same bounds.

and $\# \cdot \boldsymbol{\Sigma}^p_2$ (preferred, semi-stable, stage) for projected counting. For treewidth, many results in abstract argumentation [24], including and decomposition-guided reductions [31, 26], are known. For clique-width, [22] established dynamic programming algorithms and tractability results for acceptability under preferred semantics. SAT encodings are commonly used for solving argumentation problems [38] and QBF encodings enable tight computational bounds [36, 28]. In propositional satisfiability, tractability results for directed incidence clique-width [29], modular treewidth [42], and symmetric incidence clique-width [45] and hardness results for undirected clique-width [29] exist. Tractability results for validity of QBFs are also known for incidence treewidth and directed clique-width [8].

2 Preliminaries

We assume that the reader is familiar with standard terminology in Boolean logic [4], computational complexity [41], and parameterized complexity [16]. For an integer k, let $[k] := \{1, \ldots, k\}$ and $A \mathbin{\dot\cup} B$ be the union over disjoint sets A, B.

Satisfiability A literal is a (Boolean) variable x or its negation $\neg x$. A *clause* or *cube (also known as term)* is a finite set of literals, interpreted as the disjunction or conjunction of these literals, respectively. A *CNF formula* or *DNF formula* is a finite set of clauses, interpreted as the conjunction or disjunction of its clauses or cubes. Sometimes we say formula to refer to a CNF formula. We use the usual convention that an empty conjunction corresponds to $\top$ and an empty disjunction to $\bot$. Let φ be a formula. For a clause $c \in \varphi$, we let $\mathsf{var}(c)$ consist of all variables that occur in c and $\mathsf{var}(\varphi) := \bigcup_{c \in \varphi} \mathsf{var}(c)$. An *assignment* is a mapping $\alpha : \mathsf{var}(\varphi) \to \{0, 1\}$, α is total if it maps all variables in φ. For $x \in \mathsf{var}(\varphi)$, we define $\alpha(\neg x) := 1 - \alpha(x)$. The CNF formula φ *under the assignment* $\alpha \in 2^{\mathsf{var}(\varphi)}$ is the formula $\varphi_{|\alpha}$ obtained from φ by removing all clauses c containing a literal set to 1 by α and removing from the remaining clauses all literals set to 0 by α. An assignment α is *satisfying* if $\varphi_{|\alpha} = \emptyset$ and φ is *satisfiable* if there is a satisfying assignment α. SAT asks to decide satisfiability of φ and #SAT asks for its number of total satisfying assignments.

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. A *quantified Boolean formula* φ (in prenex

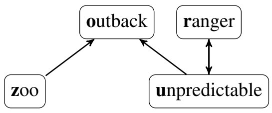

Figure 1: An example framework for deciding between going to see kangaroos in a zoo or in the outback.

normal form), *qBf* for short, is an expression of the form $\varphi = Q_1 X_1.Q_2 X_2.\cdots Q_\ell X_\ell.F(X_1,\ldots,X_\ell)$, 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 $\mathsf{matrix}(\varphi) := F$. We evaluate φ by $\exists x.\varphi \equiv \varphi[x \mapsto 1] \vee \varphi[x \mapsto 0]$ and $\forall x.\varphi \equiv \varphi[x \mapsto 1] \wedge \varphi[x \mapsto 0]$ for a variable x. W.l.o.g. we assume that $\mathsf{matrix}(\varphi) = \psi_{\text{CNF}} \wedge \psi_{\text{DNF}}$, where ψ_{CNF} is in CNF and ψ_{DNF} is in DNF. The evaluation of DNF formulas is defined in the usual way. 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 $\varphi = \exists X_1.\varphi'$ of rank ℓ, whether $\varphi \equiv 1$ holds. The problem $\#\ell$-QBF asks, given a closed qBf $\exists X_1.\varphi$ of rank ℓ, to count assignments α to X_1 such that $\varphi[\alpha] \equiv 1$. For brevity, we sometimes omit ℓ.

Computational Complexity For integer $i \geq 0$, $\exp(i, p)$ means a tower of exponentials, i.e., $\exp(i-1, 2^p)$ if $i > 0$ and p if $i = 0$. We assume that $poly(n)$ is any polynomial for given positive integer n. The Exponential Time Hypothesis (ETH) [33] is a widely accepted standard hypothesis in the fields of exact and parameterized algorithms. ETH states that there is some real $s > 0$ such that we cannot decide satisfiability of a given 3-CNF formula φ in time $2^{s\cdot|\varphi|} \cdot \|\varphi\|^{\mathcal{O}(1)}$ [16, Ch.14], where $|\varphi|$ refers to the number of variables and $\|\varphi\|$ to the *size* of φ, which is number of variables and clauses in φ.

Abstract Argumentation We use the argumentation terminology by Dung [19] and consider non-empty, finite sets A of arguments. An *argumentation framework (AF)* is a directed graph $F = (A, R)$ where A is a set of elements, called *arguments*, and $R \subseteq A \times A$, a set of pairs of arguments representing direct attacks of arguments. An argument $s \in A$ is called *defended by* S 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 argumentation, we are interested in computing so-called *extensions*, which are subsets $S \subseteq A$ of the arguments that meet certain properties according to certain semantics. We say $S \subseteq A$ is *conflict-free* if $(S \times S) \cap R = \emptyset$; S is *admissible* if (i) S is *conflict-free*, and (ii) every $s \in S$ is *defended by* S. Assume an admissible set S. Then, (iiia) S is *complete* if $\mathrm{def}_F(S) = S$; (iiib) S is *preferred*, if there is no $S' \supset S$ that is *admissible*; (iiic) S is *semi-stable* if there is no admissible set $S' \subseteq A$ with $S^+_R \subsetneq (S')^+_R$ where $S^+_R := S \cup \{a \mid (b, a) \in R, b \in S\}$; (iiid) S is *stable* if every $s \in A \setminus S$ is *attacked* by some $s' \in S$. A conflict-free set S is *stage* if there is no conflict-free set $S' \subseteq A$ with $S^+_R \subsetneq (S')^+_R$. We denote semantics by acronyms $\mathsf{adm}, \mathsf{comp}, \mathsf{pref}, \mathsf{semiSt}, \mathsf{stab}$, and stage, respectively. For a semantics $\sigma \in \{\mathsf{adm}, \mathsf{comp}, \mathsf{pref}, \mathsf{semiSt}, \mathsf{stab}, \mathsf{stage}\}$, $\sigma(F)$ is the set of *all extensions* of semantics σ in F. Given an AF $F = (A, R)$. Problem exist_σ asks whether $\sigma(F) \neq \emptyset$; $\#_\sigma$ asks for $|\sigma(F)|$; additionally given argument $a \in A$, *credulous acceptance* c_σ asks whether $a \in \bigcup_{e \in \sigma(F)} e$; and *skeptical acceptance* s_σ asks whether $a \in \bigcap_{e \in \sigma(F)} e$.

Example 2. *Consider an AF F with 4 arguments as depicted in Figure 1 arguing about watching kangaroos. Watching kangaroos in a "zoo" gives you all the excitement without needing to go the "outback". To observe them naturally, you need to see them in the "outback". However, kangaroos are "unpredictable" and can be* *dangerous. Regardless, you can go for a tour with a "ranger" making it safe to observe kangaroos in the wild, so it is not that dangerous.* $\triangleleft$

Incidence Graphs and Clique-width We follow standard terminology for graphs and directed (multi) graphs [6]. The *directed incidence graph* $\mathcal{G}_i^d(\varphi)$ of CNF (DNF) formula φ [39] is a bipartite graph with the variables and clauses (terms) of φ as vertices and a directed edge between those, indicating whether variables occur positively or negatively in a clause (term), respectively. The *incidence graph* $\mathcal{G}_i(\varphi)$ of φ omits edge directions from $\mathcal{G}_i^d(\varphi)$. We use standard definitions for clique-width [14, 12]. Intuitively, treewidth [5] measures the distance of a graph from being a tree and clique-width measures the distance of a graph to a co-graph. Clique-width is bounded by treewidth, see [10]. A graph is a *co-graph*, if it can be constructed as follows: (i) a graph with one vertex is a co-graph; for two co-graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, (iia) the disjoint union $G_1 \oplus G_2 := (V_1 \uplus V_2, E_1 \uplus E_2)$ is a co-graph; and (iib) the disjoint sum $G_1 \times G_2 := (V_1 \uplus V_2, E_1 \uplus E_2 \uplus \{\{u, v\} \mid u \in V_1, v \in V_2\})$ is a co-graph.

This lifts to k-graphs where k is a positive integer with k labels, called *colors*. The labeling of a graph $G = (V, E)$ is a function $\lambda : V \to [k]$ and a k*-graph* is a graph whose vertices are labeled by integers from $[k]$. An *initial* k*-graph* consists of exactly one vertex v colored by $c \in [k]$, denoted by $c(v)$, e.g., $1(v)$ is a shorthand for G that is a vertex v of color 1. Now, we can construct a graph G from initial k-graphs by repeatedly applying the following three *operations*. (i) *Disjoint union*, denoted by $\oplus$; (ii) *Relabeling*: changing all colors c to c', denoted by $\rho_{c \to c'}$; (iii) *Edge introduce*: connecting all vertices colored by c with all vertices colored by c', denoted by $\eta_{c,c'}$ or $\eta_{c',c}$; already existing edges are not doubled. A construction of a k-graph G using these operations can be represented by a k*-expression*, which is an algebraic term composed of $c(v)$, $\oplus$, $\rho_{c \to c'}$, and $\eta_{c,c'}$ where $c, c' \in [k]$ and v is a vertex. To construct directed graphs, $\eta_{c,c'}$ is interpreted as the operation to introduce directed edges from vertices labeled c to those labeled c'.

We describe a k-expression by a parse-tree $T = (V_T, E_T)$ and use *parse-tree of width* k synonymously. We refer by $\mathsf{chldn}(b)$ to the *set of children* of a node b in V_T. We define $\mathsf{cols} : V_T \to 2^{[k]}$ that yields the *set of colors* in the graph $G = (V, E)$ constructed *up to operation* b. Additionally, col : $(V \times V_T) \to [k]$ gives the *color of a vertex in* G *up to operation* b. The *last operation* is rt for *root*.

The *clique-width* $\mathsf{cw}(G)$ of an *undirected graph* is the smallest k such that G is definable by a k-expression. The *directed clique-width* $\mathsf{dcw}(G)$ of a *directed graph* is the smallest k such that G is definable by a k-expression. The *directed incidence clique-width* of a formula φ is $\mathsf{dcw}(\mathcal{G}_i^d(\varphi))$.

Graphs and Clique-width Let $G = (V, E)$ be a (di)graph. A graph $G' = (V', E')$ is a sub-graph of G if $V' \subseteq V$. For a set $V' \subseteq V$, the *induced sub-graph* $G[V'] = (V', E')$ consists of vertices in V' and all edges from the original graph on those vertices, i.e., $E' := \{(v_1, v_2) \mid (v_1, v_2) \in E, v_1, v_2 \in V'\}$ of edges. An *independent set* (also called stable set or co-clique) of G is a set $I \subseteq V$ in which no two vertices are adjacent. A set is independent if and only if it is a clique in the graph's complement. An independent set is a *maximal independent set (MIS)*, if there is no superset that is an independent set. A *co-graph* can be seen as a graph where any maximal clique in any of its induced subgraphs intersects any maximal independent set in that subgraph in exactly one vertex.

Directed Incidence Clique-width for CNF/DNF Let $\varphi = \{c_1, \ldots, c_m\}$ be a formula over variables X. The directed incidence graph $\mathcal{G}_i^d(\varphi)$ of φ is a bipartite graph with vertex set $X \cup \varphi$ and edge set $\{(c, x) | c \in \varphi \text{ and } x \in c\} \cup \{(x, c) | c \in \varphi \text{ and } \neg x \in c\}$. The (undirected) incidence graph $\mathcal{G}_i(\varphi)$ of φ omits the edge direction from $\mathcal{G}_i^d(\varphi)$. Observe that the orientation of edges can also be encoded by labeling them

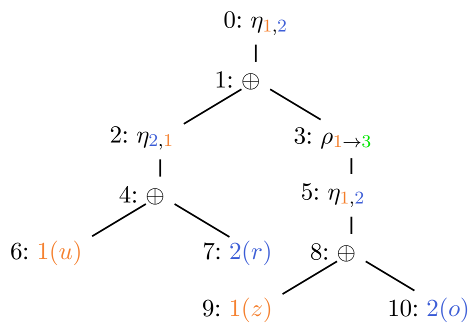

Figure 2: A k-expression of F (directed graph) from Example 2, illustrated as parse tree. We color the labels in the figure to distinguish between operation numbers and colors.

with the signs $\{+, -\}$ such that an edge between a variable x and a clause c is labeled $+$ if $x \in c$ and $-$ if $\neg x \in C$. This results in the *signed incidence graph* [29] that encodes the same information as the directed incidence graph [39]. The directed incidence graph for DNF is defined similarly after replacing clauses by terms.

Example 3 (Cont.)**.** *Consider our argumentation framework F from Example 2. The clique-width of F is ≤ 3, since only three colors are needed to draw this AF. We illustrate this in Figure 2 where concrete operations are labeled by numbers $\{0, \ldots, 10\}$ for ease and 0 refers to the root.* $\triangleleft$

Proposition 4 ([29])**.** *For any Boolean formula φ, $\mathsf{cw}(\mathcal{G}_i(\varphi)) \leq 2 \cdot \mathsf{dcw}(\mathcal{G}_i^d(\varphi))$.*

There is also a QSAT version of Proposition 1.

Proposition 5 (cf. [8])**.** *For QBFs of directed incidence clique-width k, quantifier depth ℓ, and size n, counting QSAT (on free variables) can be solved in time $\mathsf{exp}(\ell + 1, \mathcal{O}(k)) \cdot poly(n)$).*

For fixed k and an undirected graph G, one can find an $f(k)$-expression of clique-width k in polynomial time [40]; similarly for directed graphs [35].

3 K-Expression-Aware Encodings

We proceed towards reducing from argumentation problems to satisfiabilty of (quantified) Boolean formulas by a *directed decomposition-guided (DDG)* reduction that employs k-expressions. To this aim, let $F = (A, R)$ be an abstract argumentation framework from which we construct a QBF. There, we use a variable e_a for every argument $a \in A$ to store whether a is in the extension or not. We also use auxiliary variables to cover the state of attacks. In our encoding, we employ the structure and need to take care in particular of the following: (i) *initial color of arguments*, which is where we can decide whether the (single) color is defeated by being in the extension; (ii) *merge of colors*, which might occur during disjoint union or relabeling; and (iii) *edge introduce*, which occurs when drawing edges from color c' to c.

3.1 Basic Semantics

We start by presenting reductions for semantics whose classical complexity is located on the first level of the polynomial hierarchy, i.e., stable, admissible, complete extensions. For space reasons, we defer proof details to a self-archived extended version. In the following, we assume that we have a k-expression that defines the considered argumentation framework (directed graph), while requiring only k different colors. We guide our encoding along the k-expression.

<u>**Stable Extensions**</u> Here, we require conflict-freeness and that all arguments, which are not in the extension, are attacked by the extension. We encode this via a set E of *extension variables*. In the initial operation $c(a)$ for argument a of color c, the variable e_a encodes whether argument a is in the extension. Then, variable e_c^b encodes whether c contains an extension variable in operation b, which we refer to by *extension color*. To use the fact that colors have extension members, we guide the information along the k-expression. Finally, we ensure conflict-freeness of extension arguments in the (directed) edge introduction operations. Now, we construct our encoding formally.

$$e_c^b \leftrightarrow e_a \qquad \text{initial } b\text{, create } a \text{ of color } c \tag{1}$$

$$e_c^b \leftrightarrow \bigvee_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} e_c^{b'} \qquad \text{disjoint union } b\text{, every } c \in \mathsf{cols}(b) \tag{2}$$

$$e_c^b \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c' \mapsto c} e_{c'}^{b'} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b')} e_c^{b'} \qquad \text{relabeling } b\text{, every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b) \tag{3}$$

$$e_c^b \leftrightarrow e_c^{b'}, \quad \bigvee_{\text{if } b \text{ edge introduce } (c',c)} \neg e_c^b \vee \neg e_{c'}^b \qquad \text{edge intr. } b\text{, every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b) \tag{4}$$

Note that the child b' in Eq. (3) is unique. However, c could be a fresh color only introduced in b (so b' does not consider c). In this case, the disjunction would be empty (a case that is also covered here). Here, equations (1)–(4) suffice to model conflict-freeness, yielding formula $\varphi_{\#\mathsf{conf}} = \mathcal{R}_{\#\mathsf{conf} \to \#\mathrm{SAT}}(F, \mathcal{X})$. As stable extensions also attack non-extension arguments, we require for every color c that is used in an operation b an auxiliary variable d_c^b that indicates whether *every non-extension argument* of color c up to operation b is *attacked by the extension*. This is achieved via a set D of *defeated* variables, guided along the k-expression.

$$d_c^b \leftrightarrow e_c^b \qquad \text{initial } b\text{, every } c \in \mathsf{cols}(b) \tag{5}$$

$$d_c^b \leftrightarrow \bigwedge_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} d_c^{b'} \qquad \text{disjoint union } b\text{, every } c \in \mathsf{cols}(b) \tag{6}$$

$$d_c^b \leftrightarrow \bigwedge_{\text{if } b \text{ relabeling } c' \mapsto c} d_{c'}^{b'} \wedge \bigwedge_{\text{if } c \in \mathsf{cols}(b')} d_c^{b'} \qquad \text{relabeling } b\text{, every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b) \tag{7}$$

$$d_c^b \leftrightarrow d_c^{b'} \vee \bigvee_{\text{if } b \text{ edge introduce } (c',c)} e_{c'}^b \qquad \text{edge introduce } b\text{, every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b) \tag{8}$$

In the root rt, we ensure that all colors are defeated.

$$d_c^{rt} \qquad \text{for root colors } c \in \mathsf{cols}(rt) \tag{9}$$

Intuition In the initial k-graphs $b = c(a)$, we remember whether c is an extension color, this information is guided along the k-expression via other formulas. The *disjunctive* encoding, in Formulas (1)–(4), guarantees that c is an extension color if it is an extension color in some operation b. Then, Formula (1) allows to retrieve arguments from extension colors and Formula (4) requires conflict-freeness of those arguments. Moreover, given an extension color, the initial graphs uniquely determine arguments in an extension. Finally, Formulas (5)–(8) encode whether each argument is either in the extension, or attacked by it. This is again encoded via colors, where a *conjunctive* encoding is used to enforce that each argument of a color has been considered.

Example 6 (cont.). *Consider F from Example 2 and the operations corresponding to the parse tree from Figure 2. For operation* 9*, which creates the initial graph $1(z)$, we add $e_1^9 \leftrightarrow e_z$ by Formula (1) and $d_1^9 \leftrightarrow e_1^9$ by Formula (5). For operation* 8 (disjoint union)*, we add the formulas $e_1^8 \leftrightarrow e_1^9$, $e_2^8 \leftrightarrow e_2^{10}$ by Formulas (2) and $d_1^8 \leftrightarrow d_1^9$, $d_2^8 \leftrightarrow d_2^{10}$ by Formulas (6). For operation* 5 (edge-introduce)*, we obtain $e_1^5 \leftrightarrow e_1^8$, $e_2^5 \leftrightarrow e_2^8$, and $\neg e_2^5 \vee \neg e_1^5$ by Formulas (4) and $d_1^8 \leftrightarrow d_1^9$, $d_2^8 \leftrightarrow d_2^{10}$ by Formulas (8). For operation* 3 (relabeling)*, we obtain, $e_2^3 \leftrightarrow e_2^5$, $e_3^3 \leftrightarrow e_1^5$ by Formulas (3) and $d_2^3 \leftrightarrow e_2^5$, $d_3^3 \leftrightarrow e_1^5$ by Formulas (7). Rest formulas are analogous.* $\triangleleft$

From now on, we denote our reductions by *directed decomposing guided (DDG) reduction* $\mathcal{R}_{\#\sigma\rightarrow\#\mathrm{SAT}}$ and use this notion also for other semantics σ. Moreover, we denote the resulting (#)SAT-instance by $\varphi_{\#\sigma}$. First, we prove the correctness of our reduction for stable semantics.

Theorem 7 ($\star$,Correctness). *Let $F = (A, R)$ be an AF and $\mathcal{X}$ be a k-expression of F. Then, the DDG reduction $\mathcal{R}_{\#\mathsf{stab}\rightarrow\#\mathrm{SAT}}$ is correct, that is, $\#\mathsf{stab}$ on F coincides with $\#\mathrm{SAT}$ on $\mathcal{R}_{\#\mathsf{stab}\rightarrow\#\mathrm{SAT}}(F, \mathcal{X})$.*

Proof (Sketch). We establish a bijective correspondence between stable extensions of F and satisfying assignments of $\varphi_{\#\mathsf{stab}}$. In the forward direction: we construct a unique satisfying assignment α from a given stable extension S of F. This is achieved by setting the truth values for extension variables E according to S, i.e., $\alpha(e_a) = 1$ iff $a \in S$. Moreover, we set the value of $\alpha(e_c^b)$ according to the color c and operation b in the k-expression to simulate the propagation of the evaluation along the parse-tree. Then we repeat the same construction for defeated variables D. Our construction of α from S ensures that $\alpha \models \varphi_{\#\mathsf{stab}}$. To prove the uniqueness of the assignment: we observe that S uniquely determines the evaluation of α for variables $e_a \in E$, whereas the value of α for remaining variables is simply propagated due to the nature of formulas in $\varphi_{\#\mathsf{stab}}$.

In the reverse direction: we construct a unique stable extension S by considering a satisfying assignment α for $\varphi_{\#\mathsf{stab}}$. Once again, S is obtained via the extension variables in E by letting S contain an argument a iff $\alpha(e_a) = 1$. Then, it follows from Formulas 4 and 9 that S is stable. $\square$

Next, we establish that our encodings linearly preserve the clique-width. That is, reducing an AF instance to a SAT-instance increases the clique-width only linearly. Moreover, the size of the obtained formula grows with the size of the input decomposition and a cw-expression for the SAT instance does not have to be computed from scratch, given a cw-expression for the AF.

Theorem 8 ($\star$,CW-Awareness). *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{stab}\rightarrow\#\mathrm{SAT}}(F, \mathcal{X})$ constructs a SAT instance ψ that linearly preserves the directed clique-width, i.e.,* $\mathsf{dcw}(\mathcal{G}_i^d(\psi)) \in \mathcal{O}(k)$.

Proof (Sketch). Given an AF F and a k-expression $\mathcal{X}$ of F, we construct a k-expression $\mathcal{X}'$ of $\mathcal{G}_i^d(\psi)$ as follows. We first specify additional colors. For each color c, we need an *extension-version* e_c and a *defeat-version* d_c of c. Then, we additionally need two more copies for each of these versions, called *child-versions* ec_c and dc_c to add clauses corresponding to the child-operation b' of an operation b and *expired-versions* ex_c and dx_c to simulate the effect that all the edges between certain variables and their clauses have been

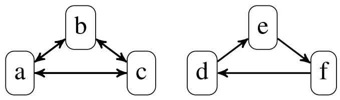

Figure 3: Example AFs highlighting the scenario where edge directions (hence resulting clique-width) matters.

added. This is required, since otherwise, we will keep adding edges from the child-versions in future, which is undesirable. Finally, to handle both positive and negative literals in clauses, we duplicate current and the child versions of each colors for $x \in \{e_c, ec_c, d_c, dc_c\}$, denoted as x^+ for positive and x^- for negative literals. For clauses, we use two additional colors, called *clause-making* (cm) and *clause-ready* (cr) to add edges between clauses and their respective literals, since we are in the setting of incidence graphs. Intuitively, when adding edges between clauses and their respective variables, we initiate a clause C to be of color cm and its literals x to be of the appropriate color x^+ or x^-. Then, we draw directed edges between C and its literals, and later relabel C to cr to avoid any further edges to/from C. Similarly, when going from an operation b' to its parent b, we change the labels of extension and defeated variables in b' to their child-versions. We conclude by observing that the above mentioned additional colors suffice to construct a k-expression $\mathcal{X}'$ of $\mathcal{G}_i^d(\psi)$, leading to a requirement of $11k + 2$ colors. □

Observe that one might be tempted to consider AFs as undirected graphs (e.g., for the stable semantics). However, this would not work as we outline below. Two AFs could be "same" considering undirected edges (or, say symmetric edges) and can thus be constructed using the same labels. However, one cannot insert directed edges using the labels for undirected edges. Thus, information is lost when encoding only undirected edges in an AF, even for stable semantics. Importantly, one can construct AFs where the directed and undirected clique-width differs from each other and such that both AFs admit different number of extensions. We next illustrate an example to further highlight this aspect.

Example 9. *Consider two AFs F_1, F_2 with arguments $\{a, b, c\}$ and $\{d, e, f\}$, respectively, together with attacks between them as depicted in Figure 3. Now, considering undirected edges (or, say symmetric edges) the two AFs are the "same". Thus, both can be constructed using the same labels. However, one cannot insert directed edges using the same labels. Observe that F_1 has a directed clique-width of two, whereas F_2 (directed cycle) has three. Importantly, F_1 admits three stable extension $\{\{a\}, \{b\}, \{c\}\}$; F_2 has none.*

Our construction relies on directed clique-width. Hence, the resulting formulas (in particular, due to Eqs. (4) and (8)) corresponding to the two AFs are also different. ◁

Admissible Extensions For admissible extensions, we require conflict-freeness and that every attack from outside is defeated. Therefore, we reuse Equations (1)–(4) to model conflict-free extensions. Furthermore, we need to maintain attacks between colors containing extension arguments. This is achieved via *attack variables* A, specified as follows.

$$\neg a^b_c \qquad \text{initial } b\text{, every } c \in \mathsf{cols}(b) \tag{10}$$

$$a^b_c \leftrightarrow \bigvee_{b' \in \mathsf{chldn}(b) : c \in \mathsf{cols}(b')} a^{b'}_c \qquad \text{disjoint union } b\text{, every } c \in \mathsf{cols}(b) \tag{11}$$

$$a^b_c \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c' \mapsto c} a^{b'}_{c'} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b')} a^{b'}_c \qquad \text{relabeling } b\text{, every } c \in \mathsf{cols}(b),$$

$$b' \in \mathsf{chldn}(b) \quad (12)$$

$$a_c^b \leftrightarrow a_c^{b'} \vee \bigvee_{\text{if } b \text{ edge introduce } (c,c')} e_{c'}^b \qquad \text{edge introduce } b\text{, every } c{\in}\mathsf{cols}(b), b'{\in}\mathsf{chldn}(b), \text{not introd. } (c', c) \quad (13)$$

$$a_c^b \leftrightarrow a_c^{b'} \wedge \neg e_{c'}^b \qquad \text{edge introduce } b\text{, every } c{\in}\mathsf{cols}(b), b' \in \mathsf{chldn}(b), \text{introd. } (c', c) \quad (14)$$

In the root rt we ensure that no color remains attacking.

$$\neg a_c^{rt} \qquad \text{for root colors } c \in \mathsf{cols}(rt) \quad (15)$$

Intuition The intuition behind this encoding is to remember whether some non-extension argument attacks an extension argument. To this aim, Formulas (10) initiate that no argument attacks an extension to begin with. Then, Formulas (11)–(12) guide this information along the k expression. Our disjunctive encoding via colors enforces that each argument of a particular color has been considered. Formulas (13)–(14) update the status of attacking argument depending on whether an argument of color c attacks an argument of the extension color c'. Finally, Formula (15) forces that no color remains attacking to our extension.

Theorem 10 ($\star$,Correctness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. Then, the DDG reduction $\mathcal{R}_{\#\mathsf{adm}\to\#\mathrm{SAT}}$ is correct, that is, $\#\mathsf{adm}$ on F coincides with $\#\mathrm{SAT}$ on $\mathcal{R}_{\#\mathsf{adm}\to\#\mathrm{SAT}}(F, \mathcal{X})$.*

Theorem 11 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{adm}\to\#\mathrm{SAT}}(F, \mathcal{X})$ constructs a SAT instance ψ that linearly preserves the width, i.e., $\mathsf{dcw}(\mathcal{G}_i^d(\psi)) \in \mathcal{O}(k)$.*

Complete Extensions For complete semantics, we compute admissible extensions such that there is no argument that could have been included. That is, there is no argument that is not in the extension but defended by it. Moreover, we additionally need the information of whether a color has been defeated or not. To achieve this, we reuse Equations (1)–(4), (10)–(14) and, on top, we compute whether we could have included an argument via a collection O of *out variables*. These are encoded and propagated as follows.

$$o_c^b \leftrightarrow \neg e_c^b \qquad \text{initial } b\text{, every } c \in \mathsf{cols}(b) \quad (16)$$

$$o_c^b \leftrightarrow \bigvee_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} o_c^{b'} \qquad \text{disjoint union } b\text{, every } c \in \mathsf{cols}(b) \quad (17)$$

$$o_c^b \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c' \mapsto c} o_{c'}^{b'} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b')} o_c^{b'} \qquad \text{relabeling } b\text{, every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b) \quad (18)$$

$$o_c^b \leftrightarrow o_c^{b'} \wedge \bigwedge_{\text{if} b \text{ edge introduce } (c,c')} \neg e_{c'}^b \qquad \text{edge introduce } b\text{, every } c{\in}\mathsf{cols}(b), b'{\in}\mathsf{chldn}(b), \text{not introd. } (c', c) \quad (19)$$

$$o_c^b \leftrightarrow o_c^{b'} \wedge (c{\neq}c') \wedge d_{c'}^{\geq b} \qquad \text{edge introduce } b\text{, every } c{\in}\mathsf{cols}(b), b'{\in}\mathsf{chldn}(b), \text{introd. } (c', c) \quad (20)$$

To compute $d_c^{\geq b}$, we need to propagate the information of being defeated backwards, from the root towards the leaves.

$$d_c^{\geq rt} \leftrightarrow \bigvee_{\text{if } rt \text{ edge introduce } (c',c)} e_{c'}^{rt} \qquad \text{for root colors } c \in \mathsf{cols}(rt) \quad (21)$$

$$d_c^{\geq b'} \leftrightarrow d_c^{\geq b} \vee \bigvee_{\text{if } b' \text{ edge introduce } (c',c)} e_{c'}^{b'} \qquad \text{non-relabeling } b \text{ with } b' \in \mathsf{chldn}(b), c \in \mathsf{cols}(b') \quad (22)$$

$$d_c^{\geq b'} \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c \mapsto c'} d_{c'}^{\geq b} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b)} d_c^{\geq b} \qquad \text{relabeling } b \text{ with } b' \in \mathsf{chldn}(b), c \in \mathsf{cols}(b') \quad (23)$$

In the root rt no color remains attacking or out.

$$\neg a_c^{rt}, \quad \neg o_c^{rt} \qquad \text{for every } c \in \mathsf{cols}(rt) \quad (24)$$

Intuition We encode whether some argument is incorrectly left out via a set O of variables. To achieve this, Formula (16) initiates arguments not in an extension as candidates for incorrectly left out. Formulas (17)–(18) guide this information along the k-expression. Our disjunctive encoding here guarantees that we have considered each argument of any color. Then, Formulas (19)–(20) update the status of an argument depending on whether there is a valid reason to leave this out. Precisely, Formula (19) guarantees that if a color c attacks an extension argument, it can not be left out since this must be defeated by the extension, due to the admissibility, and Formula (20) encodes that an argument was correctly left out, if there is some undefended attacker. Then, Formulas (24) confirm that no argument is incorrectly left out.

Theorem 12 ($\star$,Correctness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{comp}\rightarrow\#\mathrm{SAT}}$ is correct, that is, $\#\mathsf{comp}$ on F coincides with $\#\mathrm{SAT}$ on $\mathcal{R}_{\#\mathsf{comp}\rightarrow\#\mathrm{SAT}}(F, \mathcal{X})$.*

Theorem 13 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression. The reduction $\mathcal{R}_{\#\mathsf{comp}\rightarrow\#\mathrm{SAT}}(F, \mathcal{X})$ constructs a SAT instance ψ that linearly preserves the width, i.e., $\mathsf{dcw}(\mathcal{G}_i^d(\psi))) \in \mathcal{O}(k)$.*

3.2 Semantics Using Subset-Maximization

Before we turn our attention to maximization-based semantics, we require some meta-result on DNF matrices, which will substantially simplify our constructions below.

Lemma 14 ($\star$)**.** *Let Q be a QBF with inner-most $\forall$ quantifier and matrix $\varphi \wedge \psi$, where φ is in CNF and ψ is in DNF. Assuming $k{=}\,\mathsf{dcw}(\mathcal{G}_i^d(\varphi) \sqcup \mathcal{G}_i^d(\psi))$, there is a model-preserving DNF matrix φ' with $\mathsf{dcw}(\mathcal{G}_i^d(\varphi'')) \in \mathcal{O}(k)$.*

Proof (Sketch). It suffices to encode the CNF φ into a DNF formula φ' along the k-expression. The idea of our encoding is to guide the satisfiability of clauses along the k-expression, thereby keeping the information of whether a variable is assigned false or true. The resulting DNF matrix uses the formula φ' instead. Importantly, our encoding linearly preserves the directed incidence clique-width. $\square$

Preferred Extensions As before, we take an AF F, a k-expression $\mathcal{X}$ of F, and compute admissible extensions via the formula $\varphi_{\#\mathsf{adm}}$. However, we also need to keep track of subset-larger extension candidates. So, we additionally create the formula $\varphi^*_{\#\mathsf{adm}}$, where starring refers to renaming every resulting variable v by a new copy v^*. It remains to design a structure-aware encoding of subset-larger extensions via starred variables. We construct the following CNF φ_{pref} (given as Equations (25)–(31)).

$$\psi \qquad \text{for every } \psi \in \varphi^*_{\#\mathsf{adm}} \tag{25}$$

$$s_c^b \leftrightarrow e_c^{*b} \wedge \neg e_c^b \qquad \text{initial } b \text{, create } a \text{ of color } c \tag{26}$$

$$s_c^b \leftrightarrow \bigvee_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} s_c^{b'} \qquad \text{disjoint union } b \text{, every } c \in \mathsf{cols}(b) \tag{27}$$

$$s_c^b \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c' \mapsto c} s_{c'}^{b'} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b')} s_c^{b'} \qquad \text{relabeling } b \text{, every } c \in \mathsf{cols}(b),\ b' \in \mathsf{chldn}(b) \tag{28}$$

$$s_c^b \leftrightarrow s_c^{b'} \qquad \text{edge introduce } b \text{, every } c \in \mathsf{cols}(b),\ b' \in \mathsf{chldn}(b) \tag{29}$$

We skip subset-larger extension counter candidates that are not in a superset relation to the candidate.

$$e_c^b \rightarrow e_c^{*b} \qquad \text{initial } b \text{, create } a \text{ of color } c \tag{30}$$

Further, for the root operation rt, we need to find an admissible extension that is subset-maximal, expressed as follows.

$$\bigvee_{c \in \mathsf{cols}(rt)} s_c^{rt} \qquad \text{for root operation } rt \tag{31}$$

Then, the reduction $\mathcal{R}_{\#\mathsf{pref}\rightarrow\#2\text{-}\mathrm{QBF}}(F, \mathcal{X})$ constructs a QBF $\varphi_{\#\mathsf{pref}} := (\varphi_{\#\mathsf{adm}} \wedge \neg(\exists E^*, A^*, S.\varphi_{\mathsf{pref}}))$, which searches for an admissible extension (free variables) where there is no subset-larger admissible extension ($\neg\exists$). So, if there indeed is a larger extension than the candidate given via e_a^b variables, the QBF evaluates to false.

If we bring this QBF into prenex normal form (shifting negation inside), we obtain an $\forall$-QBF with free variables whose matrix is of the form $\varphi_{\#\mathsf{adm}} \wedge \varphi_{\mathsf{pref}}$ with $\varphi_{\#\mathsf{adm}}$ in CNF and φ_{pref} in DNF. This matrix can then be converted to DNF by Lemma 14. We obtain the following result.

Theorem 15 ($\star$,Correctness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{pref}\rightarrow\#2\text{-}\mathrm{QBF}}$ is correct, that is, $\#\mathsf{pref}$ on F coincides with $\#2$-QBF on $\mathcal{R}_{\#\mathsf{pref}\rightarrow\#2\text{-}\mathrm{QBF}}(F, \mathcal{X})$.*

Theorem 16 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ k-expression. The reduction $\mathcal{R}_{\#\mathsf{pref}\rightarrow\#2\text{-}\mathrm{QBF}}(F, \mathcal{X})$ constructs a QSAT instance ψ that linearly preserves the width, i.e., $\mathsf{dcw}\,\mathcal{G}_i^d(\mathsf{matrix}(\psi)) \in \mathcal{O}(k)$.*

Semi-Stable Extensions To compute semi-stable extensions, we must maximize the range of arguments. Interestingly, the range is computed, but via the information on colors we can not directly access it. We construct the following CNF $\varphi_{\mathsf{semiSt}}$ (given as equivalences).

$$\gamma \qquad \text{for every } \gamma \in \text{Formulas } (5)-(8)$$

$$\psi \qquad \text{for every } \psi \in \varphi^*_{\#\mathsf{adm}} \tag{32}$$

$$s^b_c \leftrightarrow d^{*b}_c \wedge \neg d^b_c \qquad \text{initial } b\text{, every } c \in \mathsf{cols}(b) \tag{33}$$

$$s^b_c \leftrightarrow \bigvee_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} s^{b'}_c \qquad \text{disjoint union } b\text{, every } c \in \mathsf{cols}(b) \tag{34}$$

$$s^b_c \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c' \mapsto c} s^{b'}_{c'} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b')} s^{b'}_c \qquad \text{relabeling } b\text{, every } c \in \mathsf{cols}(b),\ b' \in \mathsf{chldn}(b) \tag{35}$$

$$s^b_c \leftrightarrow (s^{b'}_c \vee d^{*b}_c) \wedge \neg d^b_c \qquad \text{edge introduce } b\text{, every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b) \tag{36}$$

For the root rt, we keep extensions of strictly larger range.

$$d^{rt}_c \rightarrow d^{*rt}_c, \quad \bigvee_{c \in \mathsf{cols}(rt)} s^{rt}_c \qquad \text{for root operation } rt \tag{37}$$

The reduction $\mathcal{R}_{\#\mathsf{semiSt}\rightarrow\#2\text{-QBF}}(F, \mathcal{X})$ constructs a QBF $(\varphi_{\#\mathsf{adm}} \wedge (\forall E^*, D^*, A^*, S.\neg\varphi_{\mathsf{semiSt}}))$, where $\neg\varphi_{\mathsf{semiSt}}$ is in DNF, but $\varphi_{\#\mathsf{adm}}$ is in CNF. As above, Lemma 14 converts the matrix into DNF as desired.

Theorem 17 ($\star$,Correctness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. Then, the DDG reduction $\mathcal{R}_{\#\mathsf{semiSt}\rightarrow\#2\text{-QBF}}$ is correct, that is, $\#\mathsf{semiSt}$ on F coincides with $\#2$-QBF on $\mathcal{R}_{\#\mathsf{semiSt}\rightarrow\#2\text{-QBF}}(F, \mathcal{X})$.*

Theorem 18 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ a k-expression. The reduction $\mathcal{R}_{\#\mathsf{semiSt}\rightarrow\#2\text{-QBF}}(F, \mathcal{X})$ constructs a QSAT instance ψ that linearly preserves the width, i.e., $\mathsf{dcw}(\mathcal{G}^d_i(\mathsf{matrix}(\psi))) \in \mathcal{O}(k)$.*

Stage Extensions The reduction $\mathcal{R}_{\#\mathsf{stage}\rightarrow\#2\text{-QBF}}(F, \mathcal{X})$ constructs a QBF $\forall E^*, D^*, S.(\varphi_{\#\mathsf{conf}} \wedge \neg\varphi_{\mathsf{stage}})$, where φ_{stage} is a CNF comprising ψ for every $\psi \in \varphi_{\#\mathsf{conf}}$ as well as Equations (33)–(37). Correctness and CW-Awareness works as above, so we obtain the following upper bounds.

Theorem 19 (Runtime-UBs)**.** *Let $F = (A, R)$ be an AF of size n and directed clique-width k, For a semantics σ, the problem $\#\sigma$ can be solved in time*

- $2^{\mathcal{O}(k)} \cdot poly(n)$ *for* $\sigma \in \{\mathsf{stab}, \mathsf{adm}, \mathsf{comp}\}$.
- $2^{2^{\mathcal{O}(k)}} \cdot poly(n)$ *for* $\sigma \in \{\mathsf{pref}, \mathsf{semiSt}, \mathsf{stage}\}$.

3.3 Credulous and Skeptical Reasoning

The preceding reductions can be extended to determine credulous and skeptical acceptance of an argument. Let F be an AF and σ be a semantics. To solve credulous acceptance c_σ for a, we append "e_a" to each formula $\varphi_{\#\sigma}$ where $e_a \in E$ is the extension variable corresponding to argument a. Then, each satisfying assignments for $\varphi_\sigma \wedge e_a$ yields an extension (via extension variables) containing a. Moreover, to solve skeptical acceptance

s_σ, we add "$\neg e_a$" to $\varphi_{\#\sigma}$ and flip the answer in polynomial time, i.e., s_σ is true for a if and only if there is no satisfying assignment for $\varphi_{\#\sigma} \wedge \neg e_a$.

We observe that correctness and CW-awareness in both cases for each semantics follows from proofs of corresponding "Correctness" and "CW-Awareness" theorems, e.g., Thm. 7 and Thm. 8 for stable semantics.

Theorem 20 (Runtime-UBs). *Let $F = (A, R)$ be an AF of size n and directed clique-width k, For a semantics σ, the problem s_σ can be solved in time*

- $2^{\mathcal{O}(k)} \cdot poly(n)$ *for* $\sigma \in \{\mathsf{stab}, \mathsf{adm}, \mathsf{comp}\}$.
- $2^{2^{\mathcal{O}(k)}} \cdot poly(n)$ *for* $\sigma \in \{\mathsf{pref}, \mathsf{semiSt}, \mathsf{stage}\}$.

c_σ *behaves similarly, but* $\mathsf{c}_{\mathsf{pref}}$ *can be solved via* $\mathsf{c}_{\mathsf{adm}}$.

4 Lower Bounds: Can We Improve?

It turns out that we can not significantly improve most of our reductions. Indeed, we can create a clique-width-aware reduction from 3SAT to asking whether some argument is included in an admissible extension.

Theorem 21 ($\star$,Admissible CW-LB). *Unless ETH fails, we can not decide for an AF $F = (A, R)$ of directed clique-width w in time $2^{o(w)} \cdot poly(|A| + |R|)$ whether there exists an admissible extension of F containing argument a.*

We can easily extend this to other semantics. Indeed, the lower bound immediately carries over to other semantics.

Corollary 22 (Stable/Complete CW-LB). *Under ETH we can not decide for an AF $F = (A, R)$ of directed clique-width w in time $2^{o(w)} \cdot poly(|A| + |R|)$ whether there is a stable/complete extension of F containing a.*

The bounds can be extended to second-level extensions.

Proposition 23 (Preferred/Semi-Stable/Stage CW-LB). *Under ETH we can not decide for an AF $F = (A, R)$ of directed clique-width w in time $2^{2^{o(w)}} \cdot poly(|A| + |R|)$ whether there exists a preferred (semi-stable/stage) extension of F that does not contain a (contains a).*

These results indicate that we cannot significantly improve our reductions. Indeed, for solving the second-level semantics, a reduction to SAT is expected to be insufficient.

5 Conclusion

Our results answer whether we can efficiently encode knowledge representation and reasoning (KRR) formalisms into (Q)SAT while respecting the clique-width. Table 1 provides a comprehensive overview. Our directed decomposition guided (DDG) reductions based on k-expressions make existing results on clique-width for SAT (Proposition 1) and QSAT (Proposition 5) accessible to abstract argumentation. Using these results and our novel DDG reduction, we establish efficient solvability when exploiting clique-width for all argumentation semantics and the problems of extension existence, acceptance, and counting. Finally, we prove that we cannot significantly improve under reasonable assumptions. Our approach remains effective even when the attack graph includes large cliques or complete bipartite structures, where other parameters (e.g., treewidth) fail.

We see various directions for future works. We are interested whether these results can be extended to other KRR formalisms such as abductive reasoning, logic-based argumentation, answer-set programming, and many more. We expect that this work might be useful for simple classroom-type proofs of Courcelle's theorem for clique-width, see [2]. Since our reductions preserve the solutions bijectively, we are interested in enumeration complexity as well. Finally, generalized or orthogonal versions of clique-width such as modular incidence treewidth and symmetric incidence clique-width.

Acknowledgment

Research was partly funded by the Austrian Science Fund (FWF), grant J4656, the French Agence nationale de la recherche (ANR), grant ANR-25-CE23-7647, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), grant TRR 318/1 2021 – 438445824, the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within project WHALE (LFN 1-04) funded under the Lamarr Fellow Network programme, and the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within project SAIL, grant NW21-059D. Part of the research was carried out while Hecher was a postdoc at MIT and while he was visiting the Simons institute for the theory of computing (part of the program *Logic and Algorithms in Database Theory and AI*). Fichte was funded by ELLIIT funded by the Swedish government.

References

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

[2] M. Bannach and M. Hecher. Structure-Guided Automated Reasoning. In *Proceedings STACS 2025*, volume 327 of *LIPIcs*, pages 15:1–15:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

[3] P. Baroni, P. E. Dunne, and G. D. Giacomo, editors. *Proceedings of COMMA'10*, Desenzano del Garda, Italy, Sept. 2010.

[4] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability – 2nd Edition*, volume 336. IOS Press, 2021.

[5] H. L. Bodlaender. Treewidth: Characterizations, applications, and computations. In F. V. Fomin, editor, *Graph-Theoretic Concepts in Computer Science*, pages 1–14, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

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

[7] R. B. Borie, R. G. Parker, and C. A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7(1):555–581, 1992.

[8] F. Capelli and S. Mengel. Tractable QBF by knowledge compilation. In *Proceedings of STACS'19*, pages 18:1–18:16. Dagstuhl Publishing, 2019.

[9] G. Charwat, W. Dvořák, S. A. Gaggl, J. P. Wallner, and S. Woltran. Methods for solving reasoning problems in abstract argumentation – a survey. *Artif. Intell.*, 220:28–63, 2015.

[10] D. G. Corneil and U. Rotics. On the relationship between clique-width and treewidth. *SIAM Journal on Computing*, 34(4):825–847, 2005.

[11] B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs* 1. *Information and computation*, 85(1):12–75, 1990.

[12] B. Courcelle. Monadic second-order logic and hypergraph orientation. In *Proceedings Eighth Annual IEEE Symposium on Logic in Computer Science*, pages 179–190, 1993.

[13] B. Courcelle. From tree-decompositions to clique-width terms. *Discr. Appl. Math.*, 248:125–144, 2018.

[14] B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46(2):218–270, 1993.

[15] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discr. Appl. Math.*, 101(1–3):77–114, 2000.

[16] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, M. P. Dániel Marx, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[17] R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, 113(1):41–85, 1999.

[18] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, London, UK, 2013.

[19] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.

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

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

[22] W. Dvorák, S. Szeider, and S. Woltran. Reasoning in argumentation frameworks of bounded clique-width. In *Proceedings of COMMA'10*, volume 216, pages 219–230. IOS Press, 2010.

[23] W. Dvořák and S. Woltran. Complexity of semi-stable and stage semantics in argumentation frameworks. *Information Processing Letters*, 110(11):425–430, 2010.

[24] W. Dvořák, R. Pichler, and S. Woltran. Towards fixed-parameter tractable algorithms for abstract argumentation. *Artif. Intell.*, 186:1–37, 2012.

[25] T. Eiter and G. Gottlob. Propositional circumscription and extended closed world reasoning are Π_2^P-complete. *Theor. Comput. Sci.*, 114(2):231–245, 1993.

[26] J. K. Fichte, M. Hecher, Y. Mahmood, and A. Meier. Decomposition-guided reductions for argumentation and treewidth. In *Proceedings of IJCAI'21*, pages 1880–1886. ijcai.org, 2021.

[27] J. K. Fichte, M. Hecher, and A. Meier. Counting complexity for reasoning in abstract argumentation. *J. Artif. Intell. Res.*, 80:805–834, 2024.

[28] J. K. Fichte, M. Hecher, and A. Pfandler. Lower Bounds for QBFs of Bounded Treewidth. In N. Kobayashi, editor, *LICS'20*, pages 410–424. ACM, 2020.

[29] E. Fischer, J. Makowsky, and E. Ravve. Counting truth assignments of formulas of bounded tree-width or clique-width. *Discr. Appl. Math.*, 156(4):511–529, 2008.

[30] E. C. Freuder. A sufficient condition for backtrack-bounded search. *J. ACM*, 32(4):755–761, Oct. 1985.

[31] M. Hecher. Treewidth-aware reductions of normal ASP to SAT - is normal ASP harder than SAT after all? In *Proceedings of KR'20*, pages 485–495, 2020.

[32] M. J. H. Heule, I. Lynce, S. Szeider, and A. Schidler. SAT encodings and beyond (dagstuhl seminar 23261). *Dagstuhl Reports*, 13(6):106–122, 2023.

[33] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[34] M. Järvisalo, T. Lehtonen, and A. Niskanen. ICCMA 2023: 5th international competition on computational models of argumentation. *Artif. Intell.*, page 104311, 2025.

[35] M. M. Kanté. The rank-width of directed graphs. *CoRR*, abs/0709.1433, 2007.

[36] M. Lampis, S. Mengel, and V. Mitsou. QBF as an alternative to Courcelle's theorem. In *Proceedings of SAT'18*, volume 10929, pages 235–252. Springer, 2018.

[37] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[38] A. Niskanen and M. Järvisalo. μ-toksia: An efficient abstract argumentation reasoner. In *Proceedings of KR'20*, pages 800–804, 2020.

[39] S. Ordyniak, D. Paulusma, and S. Szeider. Satisfiability of acyclic and almost acyclic CNF formulas. *Theor. Comput. Sci.*, 481:85–99, 2013.

[40] S. Oum and P. D. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory B*, 96(4):514–528, 2006.

[41] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[42] D. Paulusma, F. Slivovsky, and S. Szeider. Model counting for CNF formulas of bounded modular treewidth. *Algorithmica*, 76(1):168–194, 2016.

[43] A. Rago, O. Cocarascu, and F. Toni. Argumentation-based recommendations: Fantastic explanations and how to find them. In *Proceedings of IJCAI'18*, pages 1949–1955. The AAAI Press, 2018.

[44] I. Rahwan. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, July 2007.

[45] F. Slivovsky and S. Szeider. Model counting for formulas of bounded clique-width. In *Proceedings of ISAAC'13*, pages 677–687. Springer, 2013.

[46] M. Truszczynski. Trichotomy and dichotomy results on the complexity of reasoning with disjunctive logic programs. *Theory Pract. Log. Program.*, 11(6):881–904, 2011.

6 Detailed Examples

Admissible Extensions

Example 24. *Consider the AF from Example 2 and the corresponding parse tree from Figure 2. For this example, we consider the formulas added for the right side of the tree, i.e. operations* $3, 5, 8, 9, 10$. *We already discussed the formulas added for Equations (1)-(4) in Example 6. Consider now the formulas added for Equations (10)-(14). For initial operations* $9, 10$ *by Equation (10), we add the two formulas*

$$\neg a_1^9,$$
$$\neg a_2^{10}.$$

For the disjoint union operation 8, *by Equation (11), we add the formulas*

$$a_1^8 \leftrightarrow a_1^9,$$
$$a_2^8 \leftrightarrow a_2^{10}.$$

For the edge-introducing operation 5, *by Equation (13) and (14), we add the formulas*

$$a_1^5 \leftrightarrow a_1^8 \vee e_2^5,$$
$$a_2^5 \leftrightarrow a_2^8 \wedge \neg e_1^5.$$

Finally, for the relabeling operation 3, *by Equation (12), we add the formulas*

$$a_2^3 \leftrightarrow a_2^5,$$
$$a_3^3 \leftrightarrow a_1^5.$$

$\triangleleft$

Complete Extensions

Example 25. *Consider the AF from Example 2 and the corresponding parse tree from Figure 2. For this example, we consider the formulas added for the right side of the tree, i.e. operations* $3, 5, 8, 9, 10$. *We already discussed the formulas added for Equations (1)–(4) in Example 6 and those added for Equations (10)–(14) in Example 24. Consider now the formulas added for Equations (16)–(20). For initial operations* $9, 10$ *by Equation (16), we add the two formulas*

$$o_1^9 \leftrightarrow \neg e_1^9,$$
$$o_2^{10} \leftrightarrow \neg e_2^{10}.$$

For the disjoint union operation 8, *by Equation (17), we add the formulas*

$$o_1^8 \leftrightarrow o_1^9,$$
$$o_2^8 \leftrightarrow o_2^{10}.$$

For the edge-introducing operation 5, *by Equation (19) and (20), we add the formulas*

$$o_1^5 \leftrightarrow o_1^8 \wedge e_2^5,$$

$$o_2^5 \leftrightarrow o_2^8 \wedge (1 \neq 2) \wedge d_1^{\geq 5}.$$

Finally, for the relabeling operation 3*, by Equation (12), we add the formulas*

$$a_2^3 \leftrightarrow a_2^5,$$
$$a_3^3 \leftrightarrow a_1^5.$$

$\triangleleft$

Preferred Extensions

Example 26. *Consider the AF from Example 2 and the corresponding parse tree from Figure 2. For this example, we consider the formulas added for the right side of the tree, i.e. operations* $3, 5, 8, 9, 10$*. We already discussed formula* $\varphi_{\#\mathsf{adm}}$ *in Example 24. Formula* $\varphi^*_{\#\mathsf{adm}}$ *is defined analogue by renaming every variable* v *in* $\varphi_{\#\mathsf{adm}}$ *to* v^* *and thus also does not need to be discussed in this Example. We instead discuss the formulas added for Equations (26)–(29). For initial operations* $9, 10$*, by Equation (26), we add the two formulas*

$$s_1^9 \leftrightarrow e_1^{*9} \wedge \neg e_1^9,$$
$$s_2^{10} \leftrightarrow e_2^{*10} \wedge \neg e_2^{10}.$$

For the disjoint union operation 8*, by Equation (27), we add the formulas*

$$s_1^8 \leftrightarrow s_1^9,$$
$$s_2^8 \leftrightarrow s_2^{10}.$$

For the edge-introducing operation 5*, by Equation (29), we add the formulas*

$$s_1^5 \leftrightarrow s_1^8,$$
$$s_2^5 \leftrightarrow s_2^8.$$

Finally, for the relabeling operation 3*, by Equation (28), we add the formulas*

$$s_2^3 \leftrightarrow s_2^5,$$
$$s_3^3 \leftrightarrow s_1^5.$$

$\triangleleft$

7 Detailed Proofs

Stable Extensions

Theorem 7 ($\star$,Correctness)**.** *Let* $F = (A, R)$ *be an AF and* $\mathcal{X}$ *be a* k*-expression of* F*. Then, the DDG reduction* $\mathcal{R}_{\#\mathsf{stab}\to\#\mathrm{SAT}}$ *is correct, that is,* $\#\mathsf{stab}$ *on* F *coincides with* $\#\mathrm{SAT}$ *on* $\mathcal{R}_{\#\mathsf{stab}\to\#\mathrm{SAT}}(F, \mathcal{X})$*.*

Proof. "$\Longrightarrow$": Let $S \in \mathsf{stab}(F)$. We prove that $\varphi_{\#\mathsf{stab}}$ is satisfiable and the satisfying assignments coincide. To this aim, we construct a unique satisfying assignment α over variables E and D in a bottom-up fashion, as follows.

First, we set $\alpha(e_a) = 1$, and only if, $a \in S$. Then, we assign the value of α for remaining variables according to the following rules. We let $\alpha(e^b_c) = 1$ for each color c and operation b in the k-expression, if and only if,

(C1) $\alpha(e_a) = 1$
if $a \in S$, such that $b = c(a)$ is an initial k-graph for argument a colored with c;

(C2) $\alpha(e^{b'}_c) = 1$ for some $b' \in \mathsf{child}(b)$ and $c \in \mathsf{cols}(b')$,
if $b = \oplus$ and $c \in \mathsf{cols}(b)$;

(C3) either $\alpha(e^{b'}_{c'}) = 1$ or $\alpha(e^{b'}_c) = 1$,
if $b = \rho_{c' \mapsto c}$ with $c \in \mathsf{cols}(b)$ and $c' \in \mathsf{cols}(b')$;

(C4) $\alpha(e^{b'}_c) = 1$,
if $b = \eta_{c',c}$.

Furthermore, we set $\alpha(d^b_c) = 1$ for every color c and operation b of the k-expression, if and only if,

(C5) $\alpha(e^b_c) = 1$,
if $b = c(a)$ is an initial k-graph with $c \in \mathsf{cols}(b)$;

(C6) $\alpha(d^{b'}_c) = 1$ for each $b' \in \mathsf{child}(b)$ s.t. $c \in \mathsf{cols}(b')$,
if $b = \oplus$;

(C7) $\alpha(d^{b'}_{c'}) = 1$ and $\alpha(d^{b'}_c) = 1$,
if $b = \rho_{c' \mapsto c}$ with $c \in \mathsf{cols}(b) \cap \mathsf{cols}(b')$;

(C8) either $\alpha(d^{b'}_c) = 1$ or $\alpha(e^b_{c'}) = 1$,
if $b = \eta_{c',c}$ with $c \in \mathsf{cols}(b)$.

Next, we prove that α satisfies $\varphi_{\#\mathsf{stab}}$.
It is easy to observe by construction that Formulas (1)–(4a) are satisfied by α. Regarding Formulas (4b), observe that S is conflict-free. Suppose to that contrary that there is some operation $b = \eta_{c',c}$ such that $\alpha(e^b_c) = 1$ and $\alpha(e^b_{c'}) = 1$. Due to construction (C4), this is the case iff there is some child, or descendant, b' of b such that $\alpha(e^{b'}_c) = 1$ and $\alpha(e^{b'}_{c'}) = 1$ (possibly $b' \neq b$). However, the value for $\alpha(e^{b'}_c)$ for each color c and operation b' is simply propagated from some argument $a \in S$ of color c in the initial k-graph $b = c(a)$. But this is a contradiction to the conflict-freeness of S, since in some child (descendant) operation of b, there are arguments a and a' of color c and c' with $(a, a') \in R$. Therefore, either $\alpha(e^b_c) = 0$ or $\alpha(e^b_{c'}) = 0$ for every $b = \eta_{c',c}$ with $c \in \mathsf{cols}(b)$.

It can be similarly observed that α satisfies Formulas (5)–(8) by construction. It remain to prove that α satisfies Formula (9). Suppose to the contrary that there is some color c such that $\alpha(d^{rt}_c) = 0$. Due to Formulas (6)–(7), this implies that $\alpha(d^{b'}_c) = 0$ for some operation $b' \in \mathsf{chldn}(b)$ with $b = \oplus$ or $b = \rho_{c',c}$ and $c \in \mathsf{cols}(b)$. We follow such a *branch* in the k-expression via child operations $b' \in \mathsf{chldn}(b)$ where $\alpha(d^{b'}_c) = 0$. This leads us to some initial k-graph b_0 using the color c, such that $\alpha(e^{b_0}_c) = 0$ due to Formulas (5) (we prove our claim for the color c, whereas the case when c appears in a relabeling $c^* \mapsto c$ on this branch follows analogous reasoning for c^* additionally). In particular, either this branch does not contain an operation b and color c' with $c, c' \in \mathsf{cols}(b)$ such that $b = \eta_{c',c}$, or we have $\alpha(e^b_{c'}) = 0$ for such an operation $b = \eta_{c',c}$ due to Formulas (8). This implies that, there is no argument a of color c in S due to Formulas (1) and (5) and no argument of color c is attacked by any argument in S due to Formulas (8). But this leads to a contradiction, since there is at least one argument in F of color c, and every argument is either in S or attacked by S due to the stability of S. This completes our claim in this direction and hence α satisfies $\varphi_{\#\mathsf{stab}}$.

It remains to prove that α is unique for the set S. To this aim, let $\beta \neq \alpha$ be another assignment over $E \cup D$ such that β satisfies $\varphi_{\#\mathsf{stab}}$ and $\{a \mid a \in S\} = \{a \mid \beta(e_a) = 1\}$. Observe that, by definition $\beta(d^{rt}_c) = 1$ for a color c if and only if either (i) there is an edge introduction operation $b = \eta_{c'c}$ with $\beta(e^b_{c'}) = 1$, or (ii) $\beta(e^b_c) = 1$ in the initial k-graph b, or (iii) there is a relabeling operation $\rho_{c^* \to c}$ and (i) and (ii) are true for c^*. Moreover, $\beta(e^b_c) = 1$ if and only if $\beta(e_a) = 1$ for the argument a such that $b = c(a)$ is the initial operation. Since β assigns variable in E exactly as α, the same applies to variables in D. However, this leads to a contradiction and we obtain $\beta = \alpha$. Therefore, for every stable extension S in F, there is exactly one satisfying assignment α to variables $E \cup D$.

"$\Longleftarrow$": Suppose there is an assignment θ over variables in E and D such that θ satisfies $\varphi_{\#\mathsf{stab}}$. Then, we construct a stable extension as $S = \{a \mid \theta(e_a) = 1\}$ for F. Due to Formulas (9), it follows that $\theta(d^{rt}_c) = 1$ for each $c \in \mathsf{cols}(rt)$. Since θ satisfies Formulas (4) for each edge introduction operation b and color c, the set S is conflict free. Suppose to the contrary, there are arguments $a, a' \in S$ with $(a, a') \in R$. Then, let b be the edge insertion operation for colors c, c' with $col(a) = c$ and $col(a') = c'$ in b. Since, $\theta(e_a) = 1 = \theta(e_{a'})$, we have that $\theta(e^{b'}_c) = 1 = \theta(e^b_c)$ for the operation $b' \in \mathsf{chldn}(b)$. Then, due to Formulas (4a), we have that $\theta(e^b_c) = 1 = \theta(e^b_{c'})$. But this leads to a contradiction since θ can not satisfy Formulas (4a) and Formulas (4b) at the same time. Consequently, S is conflict-free.

Next, we prove that every argument $a \in A \setminus S$ is attacked by some argument $a' \in S$. Again, suppose to the contrary there is some argument $a \in A$ such that neither $a \in S$, nor is there any $a' \in S$ with $(a', a) \in R$. Let $col(a) = c$ in the root operation, i.e., the rightmost operation in the k-expression. Clearly, $\theta(d^{rt}_c) = 1$ since θ satisfies Formulas (9). From Formulas (6)–(7), we have that $\theta(d^b_c) = 1$ for every descendant b of rt such that $c \in \mathsf{cols}(b)$. Assume wlog that $col(a) = c$ in the initial operation b. The alternative case follows analogous reasoning if $col(a) = c^*$ via the relabeling $c^* \mapsto c$. This implies that we have one of the following two cases.

In the first case, we have $\theta(e^b_c) = 1$ in the $b = c(a)$ operation for creating argument a of color c due to Formula (5). But this is a contradiction to Formula (1), since $a \notin S$ and hence $\theta(e_a) = 0$. In the second case, there is some edge introduction operation b with the edge (c', c) and $\theta(e^b_{c'}) = 1$ due to Formula (8). Then, applying the same reasoning as in the only-if direction to c', we have that there is some argument $a' \in S$ due to $\theta(e_{a'}) = 1$ according to Formula (1), such that a' has color c' in b and $(a', a) \in R$, since $b = \eta_{c',c}$ introduces an edge. But this again leads to a contradiction since there is no argument $a' \in S$ attacking a. Consequently, for every satisfying assignment θ to variables $E \cup D$ of $\varphi_{\#\mathsf{stab}}$, there is exactly one stable extension S in F. This concludes correctness and establishes the claim. □

Theorem 8 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{stab}\to\#\mathrm{SAT}}(F, \mathcal{X})$ constructs a SAT instance ψ that linearly preserves the directed clique-width, i.e.,* $\mathsf{dcw}(\mathcal{G}^d_i(\psi)) \in \mathcal{O}(k)$.

Proof. Given an AF F and a k-expression $\mathcal{X}$ of F, we construct a k'-expression $\mathcal{X}'$ of $\mathcal{G}^d_i(\psi)$ as follows. We first specify additional colors needed to construct a k'-expression of ψ. For each color c, we need an *extension-version* e_c and a *defeat-version* d_c of c. Moreover, for each extension (e_c) and defeat color (d_c), we additionally need two more copies, called *child-versions* ec_c and dc_c to add clauses corresponding to the child-operation b' of a operation $b \in \mathcal{X}$ and *expired-versions* ex_c and dx_c to simulate the effect that all the edges between certain variables and their clauses have been added. This is required, since otherwise, we we will keep adding edges from the child-versions in future, which is undesirable. Finally, to handle both positive and negative literals in clauses, we replicate current and the child versions of each colors each $x \in \{e_c, ec_c, d_c, dc_c\}$ twice, denoted as x^+ for positive and x^- for negative literals. For clauses, we use two additional colors, called *clause-making* (cm) and *clause-ready* (cr) to add edges between clauses and their respective literals, since we are in the setting of incidence graphs. Intuitively, when adding edges between clauses and their respective variables, we initiate a clause C to be of color cm and its literals x to be of the appropriate color x^+ or x^-. Then, we draw directed edges between the clause C and its positive and negative literals. Once the edges for C have been added, we relabel it to cr to avoid any further edges to or from C. Similarly, when going from an operation b' to its parent b, we change the labels of extension and defeated variables in b' to their child-versions.

We first need to switch $\varphi_{\#\mathsf{stab}}$ from Formulas (1)–(9) into a collection of clauses. This can be easily achieved via iteratively replacing each formula in $\varphi_{\#\mathsf{stab}}$ using the following equivalences: (1) $\alpha \leftrightarrow \beta \equiv (\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$, (2) $\neg(x_1 \vee \cdots \vee x_n) \vee C \equiv \bigwedge_{i \leq n}(\neg x_i \vee C)$, and (3) $\neg(x_1 \wedge \cdots \wedge x_n) \vee C \equiv (\bigvee_{i \leq n} \neg x_i \vee C)$. As a result, Formulas (1)–(9) (and hence $\varphi_{\#\mathsf{stab}}$) can be converted into a CNF. Observe that, for each operation b: Formula (1) and (4a) each yields two clauses and Formula (4b) is already a clause. Likewise, for each operation b, the number for clauses corresponding to Formula (2) is bounded by the children of b via $|\mathsf{chldn}(b)|+1$, and those for Formula (3) are bounded by three (since each disjunct is just a Boolean condition). The similar argument applies to Formulas (5)–(8), thus giving a bounded many clauses in each case. This has the effect that our final expression $\mathcal{X}'$ constructed below has its size polynomial in the size of $\mathcal{X}$.

In the following, we present our construction of the expression for only one representative clause from each formula whereas analogous construction applies to the remaining cases. For each operation in the k-expression $\mathcal{X}$ of F and the corresponding clause $C \in \varphi_{\#\mathsf{stab}}$: we explain how to use additional colors described above to return a sub-expression that adds all edges between C and its literals. Intuitively, the edges for each clause from $\varphi_{\#\mathsf{stab}}$ and its corresponding literals can be added in a turn by turn fashion using auxiliary colors. Recall that we add directed edges between clauses and literal as: (C, x) if $x \in C$ and (x, C) if $\neg x \in C$ (see definition of directed incidence graph for CNF).

Each operation $b \in \mathcal{X}$ is translated into a sub-expression in $\mathcal{X}'$. For brevity, we only outline how sub-expressions corresponding to each operation are added. Precisely, we construct $\mathcal{X}'$ as follows:

- **For initial**, $b = c(a)$: we create the following sub-expression:
  1. consider the clause $C_{b,c,a} := (\neg e^b_c \vee e_a)$ corresponding to Formula (1) for the given operation b, color c, and argument a. Initiate $c^+(e_a)$, $e^-_c(e^b_c)$ and $cm(C_{b,c,a})$.
  2. take disjoint union ($\oplus$) of the operations from 1 above.
  3. add edges for the considered clause of Formula (1): η_{cm,c^+} and $\eta_{e^-_c,cm}$.
  4. relabel $C_{b,c,a}$ as completed: $\rho_{cm \mapsto cr}$.
  5. consider the clause $C_{b,c} = (d^b_c \vee \neg e^b_c)$ corresponding to Formula (5) for the given operation b and color c. Initiate $d^+_c(d^b_c)$ and $cm(C_{b,c})$.
  6. take disjoint union ($\oplus$) of the operations from 4 and 5 above.

7. add edges for the considered clause $C_{b,c}$ of Formula (5): η_{cm,d_c^+} and $\eta_{e_c^-,cm}$.
8. relabel $C_{b,c}$ as completed: $\rho_{cm\mapsto cr}$.
9. Re-iterate for any remaining clauses corresponding to Formula (1) or (5). Once all clauses have been considered, relabel extension and defeat colors to their child-version: $\rho_{e_c^o \mapsto ec_c^o}, \rho_{d_c^o \mapsto dc_c^o}$ for $o \in \{+,-\}$.

- **For disjoint union** $b = \oplus_{b' \in \mathsf{child}(b)} b'$, we create the following sub-expression:

1. consider a clause $C_{b,c}$ corresponding to Formula (2) for the given operation b and color $c \in \mathsf{cols}(b)$. Initiate $e_c^o(e_c^b), cm(C_{b,c})$ where $o = +/-$ depending on whether $C_{b,c}$ contains e_c^b or $\neg e_c^b$, respectively. Moreover, relabel (if not already) the color of $e_c^{b'}$ from the child operation b' so that we have: $ec_c^p(e_c^{b'})$ where $p = +$ if $C_{b,c}$ contains $e_c^{b'}$, and $p = -$ if it contains $\neg e_c^{b'}$.
2. take disjoint union ($\oplus$) of the operations used in 1 above.
3. add edges for the considered clause $C_{b,c}$ of Formula (2): η_{cm,e_c^o} (resp., $\eta_{e_c^o,cm}$) and $\eta_{ec_c^p,cm}$ (η_{cm,ec_c^p}) accordingly, depending on the current colors o and p for literals in 1 above.
4. relabel $C_{b,c}$ as completed: $\rho_{cm\mapsto cr}$
5. consider a clause $C_{b,c}$ corresponding to Formula (6) for given b and c. Initiate $d_c^o(d_c^b)$ and $cm(C_{b,c})$ where $o = +/-$ depending on whether $C_{b,c}$ contains d_c^b or $\neg d_c^b$, respectively. Moreover, relabel (if not already) the color of $d_c^{b'}$ from the child operation b' so that we have: $dc_c^p(d_c^{b'})$ where $p = +/-$ depending on whether $C_{b,c}$ contains $d_c^{b'}$ or $\neg d_c^{b'}$, respectively.
6. take disjoint union ($\oplus$) of operations form 4 and 5 above.
7. add edges for the considered clause $C_{b,c}$ of Formula (6): $\eta_{d_c^o,cm}$ (resp., η_{cm,d_c^o}) and η_{cm,dc_c^p} ($\eta_{dc_c^p,cm}$) accordingly, depending on the current colors o and p of literals in 6 above.
8. relabel $C_{b,c}$ as completed: $\rho_{cm\mapsto cr}$.
9. Re-iterate for any remaining clauses corresponding to Formulas (2) or (6). Once all clauses have been considered, relabel child-versions of extension and defeat colors to their expired-versions: $\rho_{ec_c^o \mapsto ex_c}, \rho_{dc_c^o \mapsto dx_c}$ for any $o \in \{+,-\}$. Relabel (current) extension and defeat colors to their child-version: $\rho_{e_c^o \mapsto ec_c^o}, \rho_{d_c^o \mapsto dc_c^o}$ for $o \in \{+,-\}$.

- **for relabeling** $b = \rho_{c' \mapsto c}$, we similarly repeat steps 1–9 as before for corresponding clauses.

1. consider a clause $C_{b,c}$ corresponding to Formula (3) for the given operation b and color $c \in \mathsf{cols}(b)$. Initiate $e_c^o(e_c^b), cm(C_{b,c})$ where $o = +/-$ depending on whether $C_{b,c}$ contains e_c^b or $\neg e_c^b$, respectively. Moreover, relabel (if not already) the color of $e_c^{b'}$ from the child operation b' so that we have: $ec_c^p(e_c^{b'})$ where $p = +$ if $C_{b,c}$ contains e_c^b, and $p = -$ if it contains $\neg e_c^b$.
2. take disjoint union ($\oplus$) of operation used in 1 above.
3. add edges for the considered clause $C_{b,c}$ of Formula (3): $\eta_{e_c^o,cm}$ (resp., η_{cm,e_c^o}) and η_{cm,ec_c^p} ($\eta_{ec_c^p,cm}$) accordingly, depending on the current colors o and p of literals in 1 above.
4. relabel $C_{b,c}$ as completed: $\rho_{cm\mapsto cr}$

5. consider a clause $C_{b,c}$ corresponding to Formula (7) for given b and c. Initiate $d^o_c(d^b_c)$ and $cm(C_{b,c})$ where $o = +/-$ depending on whether $C_{b,c}$ contains d^b_c or $\neg d^b_c$, respectively. Moreover, relabel (if not already) the color of $d^{b'}_c$ from the child operation b' so that we have: $dc^p_c(d^{b'}_c)$ where $p = +/-$ depending on whether $C_{b,c}$ contains $d^{b'}_c$ or $\neg d^{b'}_c$, respectively.
6. take disjoint union ($\oplus$) of operations form 4 and 5 above.
7. add edges for the considered clause $C_{b,c}$ of Formula (7): $\eta_{d^o_c,cm}$ (resp., η_{cm,d^o_c}) and η_{cm,dc^p_c} ($\eta_{dc^p_c,cm}$) accordingly, depending on the current colors o and p of literals in 6 above.
8. relabel $C_{b,c}$ as completed: $\rho_{cm\mapsto cr}$.
9. Re-iterate for any remaining clauses corresponding to Formulas (3) or (7). Once all clauses have been considered, relabel child-versions of extension and defeat colors to their expired-versions: $\rho_{ec^o_c\mapsto ex_c}, \rho_{dc^o_c\mapsto dx_c}$ for any $o \in \{+,-\}$. Relabel (current) extension and defeat colors to their child-version: $\rho_{e^o_c\mapsto ec^o_c}, \rho_{d^o_c\mapsto dc^o_c}$ for $o \in \{+,-\}$.

- **for edge introduction** $b = \eta_{c',c}$: we again repeat steps.

1. consider a clause $C_{b,c}$ corresponding to Formula (4) for the given operation b and color $c \in \mathsf{cols}(b)$. Initiate $e^o_c(e^b_c), cm(C_{b,c})$ where $o = +/-$ depending on whether $C_{b,c}$ contains e^b_c or $\neg e^b_c$, respectively. Moreover, relabel (if not already) the color of $e^{b'}_c$ from the child operation b' so that we have: $ec^p_c(e^{b'}_c)$ where $p = +$ if $C_{b,c}$ contains e^b_c, and $p = -$ if it contains $\neg e^b_c$.
2. take disjoint union ($\oplus$) of operation used in 1 above.
3. add edges for the considered clause $C_{b,c}$ of Formula (4): $\eta_{e^o_c,cm}$ (resp., η_{cm,e^o_c}) and η_{cm,ec^p_c} ($\eta_{ec^p_c,cm}$) accordingly, depending on the current colors o and p of literals in 1 above.
4. relabel $C_{b,c}$ as completed: $\rho_{cm\mapsto cr}$
5. consider a clause $C_{b,c}$ corresponding to Formula (8) for given b and c. Initiate $d^o_c(d^b_c)$ and $cm(C_{b,c})$ where $o = +/-$ depending on whether $C_{b,c}$ contains d^b_c or $\neg d^b_c$, respectively. Moreover, relabel (if not already) the color of $d^{b'}_c$ from the child operation b' so that we have: $dc^p_c(d^{b'}_c)$ where $p = +/-$ depending on whether $C_{b,c}$ contains $d^{b'}_c$ or $\neg d^{b'}_c$, respectively. In this case, we also need to consider variables $e^b_{c'}$. Thus, we relabel (if not already) the color of $e^b_{c'}$ so that we have: $e^p_{c'}(e^b_{c'})$ where $p = +/-$ depending on whether $C_{b,c}$ contains $e^b_{c'}$ or $\neg e^b_{c'}$, respectively
6. take disjoint union ($\oplus$) of operations form 4 and 5 above.
7. add edges for the considered clause $C_{b,c}$ of Formula (8): $\eta_{d^o_c,cm}$ (resp., η_{cm,d^o_c}) and η_{cm,dc^p_c} ($\eta_{dc^p_c,cm}$) accordingly, depending on the current colors o and p of literals in 6 above.
8. relabel $C_{b,c}$ as completed: $\rho_{cm\mapsto cr}$.
9. Re-iterate for any remaining clauses corresponding to Formulas (4) or (8). Once all clauses have been considered, relabel child-versions of extension and defeat colors to their expired-versions: $\rho_{ec^o_c\mapsto ex_c}, \rho_{dc^o_c\mapsto dx_c}$ for any $o \in \{+,-\}$. Relabel (current) extension and defeat colors to their child-version: $\rho_{e^o_c\mapsto ec^o_c}, \rho_{d^o_c\mapsto dc^o_c}$ for $o \in \{+,-\}$.

- In the root operation, we only need to initiate clauses C_c due to Formulas (9) and add the corresponding single edges from these clauses to d_c.

The correctness of the k'-expression follows from the fact that (I.) each clause from $\varphi_{\#\mathsf{stab}}$ is eventually added, (II.) only edges due to the clauses in $\varphi_{\#\mathsf{stab}}$ are added, and (III.) the direction of each edge correctly depicts the membership of a literal in the clause. Observe that (III.) holds since we add an edge $C{\rightarrow}x$ if $x \in C$ and $C{\leftarrow}x$ if $\neg x \in C$. Corresponding to each color c, we use five copies used as extension versions, given as $\{e_c^+, e_c^-\} \cup \{ec_c^+, ec_c^-\} \cup \{ex_c\}$. Likewise, we use five copies of defeat versions of each color. Moreover, two additional colors are required for adding edges between clauses and literals. We conclude by observing that, one requires $5k + 5k$ additional colors for drawing the incidence graph of the formula. This results in a requirement of $k' = 11k + 2$ colors to add the expression $\mathcal{X}'$, hence an increase of clique-width which is still linear in k. □

Admissible Extensions

Theorem 10 ($\star$,Correctness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. Then, the DDG reduction $\mathcal{R}_{\#\mathsf{adm}\rightarrow\#\mathrm{SAT}}$ is correct, that is, $\#\mathsf{adm}$ on F coincides with $\#\mathrm{SAT}$ on $\mathcal{R}_{\#\mathsf{adm}\rightarrow\#\mathrm{SAT}}(F, \mathcal{X})$.*

Proof. "$\Longrightarrow$": Let $S \in \mathsf{adm}(F)$. As before, we prove that $\varphi_{\#\mathsf{adm}}$ is satisfiable by constructing an assignment α over variables E and A in a bottom-up fashion. The evaluation for variables in E remains as in the proof of Theorem 7 before.

Regarding the attacking variables, we construct α as follows. For every color c and operation b of the k-expression, we set $\alpha(a_c^b) = 0$ iff either (A1) b is an initial operation with $c \in \mathsf{cols}(b)$; or (A2) $\alpha(a_c^{b'}) = 0$ for each $b' \in \mathsf{child}(b)$ s.t. $c \in \mathsf{cols}(b')$, if b is disjoint union operation; or (A3) $\alpha(a_{c'}^{b'}) = 0$ and $\alpha(a_c^{b'}) = 0$, if $b = \rho_{c'\mapsto c}$ with $c \in \mathsf{cols}(b) \cap \mathsf{cols}(b')$; or (A4a) $\alpha(a_c^{b'}) = 0$ and $\alpha(e_{c'}^b) = 0$, if $b = \eta_{c,c'}$ but not $\eta_{c',c}$ with $c \in \mathsf{cols}(b)$; (A4b) either $\alpha(a_c^{b'}) = 0$ or $\alpha(e_{c'}^b) = 0$, if $b = \eta_{c',c}$ with $c \in \mathsf{cols}(b)$.

We argue that α satisfies $\varphi_{\#\mathsf{adm}}$. Observe that Formulas (1)–(4) are satisfied by α due to the proof of Theorem 7 and the fact that S is conflict-free. Similarly, α satisfies Formulas (10)–(14) by construction. It remains to prove that α satisfies Formulas (15) for each color c. Suppose to the contrary that there is some color c such that $\alpha(a_c^{rt}) = 1$. Observe that $\alpha(a_c^b) = 0$ for each initial operation b initiating the color c (due to Formulas (10)). Moreover, Formulas (11)–(12) then merely propagate the values from child operation to their parents. However, $\alpha(a_c^{rt}) = 1$ implies that the value for $\alpha(a_c^b)$ changes at some edge introduction operation b due to Formulas (13). In particular, $\alpha(a_c^b) = 1$ for at least one such operation b. Due to Formulas (13), either $\alpha(a_c^{b'}) = 1$ for $b' \in \mathsf{chldn}(b)$, or $\alpha(e_{c'}^b) = 1$ if $b = \eta_{c,c'}$. We consider the latter case, as the former case leads to repeating the same argument for $a_c^{b'}$. Since $\alpha(e_{c'}^b) = 1$, this implies that c' is an extension color in b, hence there is some argument a' of color c' such that $a' \in S$. Moreover, due to the edge introduction (c, c'), there is an argument a of color c such that $(a, a') \in R$. Since S is admissible, a' must be defended against a, i.e., there is some argument $a'' \in S$ with $(a'', a) \in R$. Suppose a'' be of color c''. Due to $(a'', a) \in R$, there is an edge introduction operation b'' such that Formula (14) applies to c for introducing the edge (c'', c). This implies that $\alpha(a_c^{b''}) = 0$ since $\alpha(e_{c''}^{b''}) = 1$ due to Formulas (14) for b'' and c. Since b was an arbitrary operation with $\alpha(a_c^b) = 1$, this implies that for each such b, we can find such an edge introduction operation due to the admissibility of S. Hence, we obtain a contradiction to $\alpha(a_c^b) = 1$ and hence $\alpha(a_c^b) = 0$ for any operation b, in particular for $b = rt$. This completes our claim in this direction. Hence, α satisfies $\varphi_{\#\mathsf{adm}}$.

To prove that α is unique for the set S, we again consider another assignment $\beta \neq \alpha$ over $E \cup A$ that agrees with α over extension variables. However, as before, since β assigns variable in E exactly as α, the same applies to variables in A since β has to satisfy Formulas (16) and (24). This leads to a contradiction. Hence, for every admissible extension S in F, there is exactly one satisfying assignment α to variables $E \cup A$.

"$\Longleftarrow$": Suppose there is an assignment θ over variables E and A such that $\theta \models \varphi_{\#\mathsf{adm}}$. Then, we construct an admissible extension $S = \{a \mid \theta(e_a) = 1\}$ for F. S is conflict-free due to Theorem 7 since θ satisfies Formulas (1)–(4). To prove admissibility, let $a' \in S$ be an argument. Suppose to the contrary, there is an argument $a \in A \setminus S$ such that $(a, a') \in R$, but there is no $a'' \in S$ with $(a'', a) \in R$. Due to Formulas (15), it follows that $\theta(a_c^{rt}) = 0$ for each $c \in \mathsf{cols}(rt)$. In particular, $\theta(a_c^{rt}) = 0$ for the color c of argument a in the root operation of the k-expression. We prove that this is not possible since the argument a with $col(a) = c$ still attacks our extension S. Since $(a, a') \in R$, there is an edge introduction operation $b = \eta_{c,c'}$, wlog, since the similar argument applies if this happens before one of the colors were relabeled to c or c'. However, then we have $\theta(e_{c'}^b) = 1$ since $a' \in S$. But then, this leads to $\theta(a_c^b) = 1$ due to Formulas (13) since the edge $\eta_{c,c'}$ is being introduced instead of $\eta_{c',c}$. Moreover, there is no $a'' \in S$ attacking a therefore Formula (14) does not apply to the color c of a in any operation. Consequently, $\theta(a_c^b) = 1$ is propagated until the root operation and hence $\theta(a_c^{rt}) = 1$. However, this leads to a contradiction since θ satisfies Formulas (15) for each c.
This concludes the claim and completes the correctness. □

Theorem 11 ($\star$,CW-Awareness). *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{adm}\to\#\mathrm{SAT}}(F, \mathcal{X})$ constructs a SAT instance ψ that linearly preserves the width, i.e.,* $\mathsf{dcw}(\mathcal{G}_i^d(\psi)) \in \mathcal{O}(k)$.

Proof. The proof follows analogous reasoning to the proof of Theorem 8. This is established by (1) replacing variables in D by those in A, (2) renaming *defeat* versions of each color (d_c) to be attack color (a_c) to fit our intuition, and (3) adding edges corresponding to clauses arising from Formulas (10)–(15) instead of Formulas (5)–(9). It is easy to observe that one still needs $11k + 2$ colors. As a result, the number of colors still increases only linearly. □

Complete Extensions

Theorem 12 ($\star$,Correctness). *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{comp}\to\#\mathrm{SAT}}$ is correct, that is, $\#\mathsf{comp}$ on F coincides with $\#\mathrm{SAT}$ on $\mathcal{R}_{\#\mathsf{comp}\to\#\mathrm{SAT}}(F, \mathcal{X})$.*

Proof. "$\Longrightarrow$": Let S be a complete extension in F. We prove that $\varphi_{\#\mathsf{comp}}$ is satisfiable by constructing an assignment α over variables E, A, O and D in a bottom-up fashion. The evaluation for variables in E and A remains the same as in the proof of Theorem 10.

Regarding the out O and defeated D variables, we construct α as follows. First, we set $\alpha(d_c^{\geq b'}) = 1$ iff (D1) either $\alpha(d_c^{\geq b}) = 1$ or $\alpha(e_{c'}^b) = 1$ for any $b \neq \rho$ with $b' \in \mathsf{chldn}(b)$ and $c \in \mathsf{cols}(b')$; or (D2) $\alpha(d_{c'}^{\geq b}) = 1$ if $b = \rho_{c \mapsto c'}$ for $b' \in \mathsf{chldn}(b)$ and $c \in \mathsf{cols}(b)$, or if $\alpha(d_c^{\geq b}) = 1$ for the non-relabeling c. Then, for every color c and operation b of the k-expression, we set $\alpha(o_c^b) = 1$ iff either (O1) $\alpha(e_c^b) = 0$ for b an initial operation with $c \in \mathsf{cols}(b)$; or (O2) $\alpha(o_c^{b'}) = 1$ for some $b' \in \mathsf{child}(b)$ s.t. $c \in \mathsf{cols}(b')$, if $b = \oplus$; or (O3) either $\alpha(o_c^{b'}) = 1$ if $c \in \mathsf{cols}(b')$ or $\alpha(o_{c'}^{b'}) = 1$, for $b = \rho_{c' \mapsto c}$ with $c \in \mathsf{cols}(b)$; or (O4a) $\alpha(o_c^{b'}) = 1$ and $\alpha(o_{c'}^b) = 0$, if $b = \eta_{c,c'}$ but not $\eta_{c',c}$ with $c \in \mathsf{cols}(b)$; (O4b) $\alpha(o_c^{b'}) = 1$ with $c \neq c'$ and additionally $\alpha(d_{c'}^{\geq b}) = 1$, if $b = \eta_{c',c}$ with $c \in \mathsf{cols}(b)$.

We argue that $\alpha \models \varphi_{\mathsf{comp}}$. Observe that Formulas (1)–(4) are satisfied by α due to the proof of Theorem 7 and the fact that S is conflict-free whereas Formulas (10)–(14) due to the proof of Theorem 10 and admissibility of S. Moreover, α satisfies Formulas (16)–(23) by construction. It remain to prove that α satisfies Formulas (24) for each color c. The case for $\neg a_c^{rt}$ follows due to the proof of Theorem 10 since S is admissible. Therefore, we consider the only remaining case of $\neg o_c^{rt}$. To this aim, suppose to the contrary that there is some color c such that $\alpha(o_c^{rt}) = 1$.

Observe that this is only possible if $\alpha(e^b_c) = 0$ for the initial operation b initiating some argument a of color c (due to Formulas (16)). This is without loss of generality since the case of a different color $c^* \neq c$ follows analogous reasoning due to the relabeling in Formulas (18). Since $\alpha(o^{rt}_c) = 1$, this implies that the value for $\alpha(o^b_c)$ must not change at any edge introduction operation b due to Formulas (19)–(20). Now, we look at the acceptance status of the argument a with respect to S. Since $\alpha(e^b_c) = 0$, we have that $\alpha(e_a) = 0$ due to Formula (1). Intuitively, a is not in the extension but has been incorrectly left out. To obtain a contradiction, we prove that a is defended by S. If a is not attacked by any argument in F, then we get a contradiction straightforwardly since $a \in S$ must be true as S is complete. Now, let $a' \in A$ be an arbitrary argument such that $(a', a) \in R$. Moreover, let $col(a') = c'$ in the operation b when the attack (a', a) is added. We consider the following two cases.

(I). $a' \in S$. Then, a can not be defended by S since this would mean that there exists $s' \in S$ with $(s', a') \in R$, being a contradiction to S being conflict-free. But this implies that $\alpha(d^b_{c'}) = 0$ due to Formula (22) for the color c' of a' in b. However, this leads to a contradiction due to Formula (20), since $\alpha(o^{b'}_c) = 0$ for every ancestor operation b' of b, and in particular $\alpha(o^{rt}_c) = 0$.

(II). $a' \notin S$. Then, we again consider two sub-cases depending on whether a' is attacked or not by S. The scenario when a' is not attacked by S follows the same reasoning as in case (I), since c' is not defeated and hence $\alpha(o^{rt}_c) = 0$ leading to a contradiction. In the second scenario, if a' is attacked by S, then $\alpha(d^{\geq b}_{c'}) = 1$ (due to Formula (22) since there is some argument $s \in S$ of color c'' and an edge introduction operation (c'', c)) and hence $\alpha(o^b_c) = 1$ still remains true due to Formula (20). However, since a' was arbitrary, this is true for any $a' \in A$. As a result, the argument a is actually defended by S, and hence $a \in S$ must be true since S is a complete extension. But this leads to a contradiction to the fact that $\alpha(e_a) = 0$ due to Formula (1) and (16). This completes our claim in this direction and hence $\alpha \models \varphi_{\mathsf{comp}}$.

The argument that α is unique for the set S follows similar reasoning as in the proof of Theorem 10. Consider another assignment $\beta \neq \alpha$ over $E \cup A \cup O$ that agrees with α over extension variables. As before, since β sets variable in E exactly as α, the same applies to variables in A (due to satisfaction of Formulas (16) and (24)) and O (due to Formulas (16) and (24)). This leads to a contradiction and hence, for every complete extension S in F, there is exactly one satisfying assignment α to variables $E \cup A$.

"$\Longleftarrow$": Suppose there is an assignment θ over variables E, A, O and D such that $\theta \models \varphi_{\#\mathsf{comp}}$. Then, we construct a complete extension $S = \{a \mid \theta(e_a) = 1\}$ for F. S is clearly conflict free due to Theorem 7 since θ satisfies Formulas (1)–(4). Moreover, S is admissible due to the proof of Theorem 10 and Formulas (10)–(15). To prove that S is in fact compete, suppose to the contrary, there is an argument $a \in A \setminus S$ such that for every $(a', a) \in R$, there is some $s \in S$ with $(s, a') \in R$. Due to Formulas (24), it follows that $\theta(o^{rt}_c) = 0$ for each $c \in \mathsf{cols}(rt)$, in particular, for the color c of argument a. We prove that this is not possible since the argument a with $col(a) = c$ is incorrectly left out from S.

Since $a \notin S$, we have $\theta(e_a) = 0$ as well as $\theta(e^b_c) = 0$ due to Formula (1) for the initial operation b for argument a and color c. Hence, $\theta(o^b_c) = 1$ due to Formula (16) for the same operation b and color c. Since there is an argument a' with $(a', a) \in R$, we look at the satisfaction of Formulas (20) for the edge introduction operation $b = \eta_{c',c}$ where a' has color c' in b. Since, $(s, a') \in R$ for each attacker a' of a, we have that $\theta(e^{b_s}_{c_s}) = 1$ for some edge introduction operation b_s with $\mathsf{cols}(s) = c_s$ in b_s. As a result, we have $\theta(d^{\geq b}_{c'}) = 1$ due to Formula (22)–(23). But this implies that, $\theta(o^b_c) = 1$ is true for this edge introduction operation b, and also propagated to every ancestor operation of b. We next consider the following two cases when this value can change, and prove that both lead to a contradiction.

(I). In some edge introduction operation b, Formula (19) applies. Then, due to the edge introduction $b = \eta_{c,c'}$, we have $\theta(o^b_c) = 0$ iff $\theta(e^b_{c'}) = 1$. That is, a attacks an argument s' of color c' and $s' \in S$. But this leads to a contradiction as S can not defend a, being a conflict-free set.

(II). In some edge introduction operation b, Formula (20) applies. Then, due to the edge introduction $b = \eta_{c',c}$, we have $\theta(o_c^b) = 0$ iff $\theta(d_{c'}^{\geq b}) = 0$. In other words, a is attacked by an argument a' of color c' but not defended against it. This again leads to a contradiction since S defends a as per our assumption.

As a result, we must have $\theta(o_c^{rt}) = 1$. However, this contradicts to the fact that $\theta \models \varphi_{\#\mathsf{comp}}$ (in particular, due to Formula (24)).

This concludes correctness and establishes the claim. $\square$

Theorem 13 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression. The reduction $\mathcal{R}_{\#\mathsf{comp}\to\#\mathrm{SAT}}(F, \mathcal{X})$ constructs a SAT instance ψ that linearly preserves the width, i.e., $\mathsf{dcw}(\mathcal{G}_i^d(\psi))) \in \mathcal{O}(k)$.*

Proof. Following analogous reasoning to the proof of Theorem 11, we highlight the changes due to Formulas (16)–(21). This is established by adding (1) a copy of variables in A renamed to serve 'out' variables O, (2) having *out* versions of each colors (o_c) corresponding to each version of the attack color (a_c), and (3) adding edges corresponding to clauses arising from Formulas (16)–(24) by simply adapting the same procedure as for Formulas (10)–(15). Finally, for each operation we need to add clauses corresponding to Formulas (21)–(22), which can be achieved by copying their clauses accordingly and using additional colors to model defeat variables. Observe that the additional colors are needed to add clauses for out and defeat variables. It can be noticed that the number of colors required now doubles, compared to the case of admissible semantics. The additional $11k + 2$ colors are needed due to the positive/negative and current/child/expired versions of attack and defeat colors used in the expression construction of $\varphi_{\#\mathsf{comp}}$. As a result, the total number of colors being $2 \cdot (11k + 2)$ still increases only linearly. $\square$

Meta Result on DNF Matrices

Lemma 27 ($\star$)**.** *Let Q be a QBF with inner-most $\forall$ quantifier and matrix $\varphi \wedge \psi$, where φ is in CNF and ψ is in DNF. Assuming $k{=}\mathsf{dcw}(\mathcal{G}_i^d(\varphi) \sqcup \mathcal{G}_i^d(\psi))$, there is a model-preserving DNF matrix φ' with $\mathsf{dcw}(\mathcal{G}_i^d(\varphi'')) \in \mathcal{O}(k)$.*

Proof. It suffices to encode the CNF φ into a DNF formula along the k-expression. To this end, we require (universally quantified) auxiliary variables of the form sat_c^b, indicating that φ is satisfied for color c up to b as well as t_c^b or f_c^b, which indicates that c in b has a variable that is true or false, respectively. We construct DNF $\neg\varphi'$ with φ' given next:

$$sat_c^b,\ \ t_c^b \leftrightarrow v,\ \ f_c^b \leftrightarrow \neg v \qquad \text{initial } b\text{, variable } v \text{ of color } c \quad (38)$$

$$\neg sat_c^b,\quad \neg t_c^b,\quad \neg f_c^b \qquad \text{initial } b\text{, clause } f \text{ of color } c \quad (39)$$

$$sat_c^b \leftrightarrow \bigwedge_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} sat_c^{b'}, \qquad \text{disjoint union } b\text{, every } c \in \mathsf{cols}(b)$$

$$t_c^b \leftrightarrow \bigvee_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} t_c^{b'},\qquad f_c^b \leftrightarrow \bigvee_{b' \in \mathsf{chldn}(b): c \in \mathsf{cols}(b')} f_c^{b'} \quad (40)$$

$$sat_c^b \leftrightarrow \bigwedge_{\text{if } b \text{ relabeling } c' \mapsto c} sat_{c'}^{b'} \wedge \bigwedge_{\text{if } c \in \mathsf{cols}(b')} sat_c^{b'}, \qquad \text{relabeling } b\text{, every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b)$$

$$t_c^b \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c' \mapsto c} t_{c'}^{b'} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b')} t_c^{b'},\quad f_c^b \leftrightarrow \bigvee_{\text{if } b \text{ relabeling } c' \mapsto c} f_{c'}^{b'} \vee \bigvee_{\text{if } c \in \mathsf{cols}(b')} f_c^{b'} \quad (41)$$

$$sat_c^b \leftrightarrow sat_c^{b'} \vee \bigvee_{\substack{\text{if } b \text{ +edge} \\ \text{introduce } (c', c)}} t_{c'}^b \vee \bigvee_{\substack{\text{if } b \text{ −edge} \\ \text{introduce } (c', c)}} f_{c'}^b, \qquad \text{edge introduce } b, \text{ every } c \in \mathsf{cols}(b), b' \in \mathsf{chldn}(b)$$

$$t_c^b \leftrightarrow t_c^{b'}, \quad f_c^b \leftrightarrow f_c^{b'} \tag{42}$$

In the end, we require satisfiability for every color:

$$sat_c^{rt} \qquad \text{for root colors } c \in \mathsf{cols}(rt) \tag{43}$$

The idea of this approach is to guide satisfiability of clauses (sat_c^b) along, thereby keeping the information of whether a variable is assigned false (f_c^b) or true (t_c^b). Then, the resulting DNF matrix will be $\varphi'' = \neg\varphi' \wedge \psi$. It is easy to see that the directed incidence clique-width is linearly preserved. ☐

Preferred Extensions

Theorem 15 ($\star$,Correctness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. The DDG reduction $\mathcal{R}_{\#\mathsf{pref}\to\#2\text{-}\mathrm{QBF}}$ is correct, that is, $\#\mathsf{pref}$ on F coincides with $\#2$-QBF on $\mathcal{R}_{\#\mathsf{pref}\to\#2\text{-}\mathrm{QBF}}(F, \mathcal{X})$.*

Proof. Observe that, following Theorem 10, there exists a one-one correspondence between admissible extension S in F and satisfying assignments α for $\varphi_{\#\mathsf{adm}}$. Therefore, it suffices to prove that the Formulas (26)–(31) only allow subset-maximal admissible (hence preferred) extensions.

"$\Longrightarrow$" Let S be a preferred extension in F. Then, we construct a unique satisfying assignment α for $\varphi_{\#\mathsf{pref}}$. We set, $\alpha(e_a) = 1$ for every $a \in S$ and $\alpha(e_a) = 0$ otherwise. Furthermore, for remaining variables in E and A, we map them exactly as in the proof of Theorem 10. Then, $\alpha \models \varphi_{\#\mathsf{adm}}$ since S is admissible.

Now, we prove that $\varphi_{\#\mathsf{pref}}[\alpha] \equiv 1$. Assume towards a contradiction that there exists an assignment β over variables in E^*, A^*, S, such that $\varphi_{\#\mathsf{pref}}[\alpha][\beta] \equiv 1$. In particular, this implies that $\beta \models \varphi^*_{\#\mathsf{adm}}$ due to Formulas (25). Moreover, there is some color c such that $\beta(s_c^{rt}) = 1$. due to Formulas 31. However, this implies that there is some initial operation b such that $\beta(s_{c'}^b) = 1$ for some color (possibly) $c' = c$ by Formulas (27)–(29). But this implies that $\beta(e_c^{*b}) = 1$ and $\alpha(e_c^b) = 0$ must be true. We construct an admissible set S^* via the assignment β to variables in E^*. That is, we let $S^* = \{a \mid a \in F, \beta(e_a^*) = 1\}$. Then S is indeed admissible since β satisfies $\varphi^*_{\#\mathsf{adm}}$ due to Formulas (25). Moreover, since $\varphi_{\#\mathsf{pref}}[\alpha][\beta] \equiv 1$, the two assignments together satisfy Formulas (30) and hence $S \subseteq S^* \subseteq A$ by definition of S^* and α. Furthermore, we have $S \subsetneq S^*$ due to Formula (26). However, this leads to a contradiction to S being preferred since S^* is also admissible. Consequently, there is no such assignment β satisfying Formulas (26)–(31). Finally, observe that there is no assignment α' over variables $E \cup A$ such that $\alpha' \neq \alpha$ and α' is a satisfying assignment for $\varphi_{\#\mathsf{pref}}$ if $\{a \mid a \in S\} = \{a \mid \alpha'(e_a) = 1\}$. This holds due to the same claim for $\varphi_{\#\mathsf{adm}}$, since there is bijection between admissible extensions and satisfying assignments for $\varphi_{\#\mathsf{adm}}$.

"$\Longleftarrow$". Let α be a satisfying assignment over $E \cup A$ such that $\varphi_{\#\mathsf{pref}}[\alpha] \equiv 1$. Then, we construct an extension $S = \{a \mid a \in A, \alpha(e_a) = 1\}$ and prove that S is a preferred in F. S is indeed admissible since $\alpha \models \varphi_{\#\mathsf{adm}}$. Now, assume towards a contradiction and let S' be another admissible set in F such that $S' \supset S$. Then, we construct an assignment β using S' such that $\varphi_{\#\mathsf{pref}}[\alpha][\beta] \equiv 1$. For $a \in A$, we define $\beta(s_c^b) = 1$ if there is an argument $a \in A$ in the leaf operation of our k-expression with color c such that $a \in S' \setminus S$, and we set $\beta(s_c^b) = 0$, otherwise. Furthermore, we for each non-leaf b, we set $\beta(s_c^b) = 1$ iff either (1) $\beta(s_c^{b'}) = 1$ for $b' \in \mathsf{child}(b)$, if b is disjoint union, or (2) either $\beta(s_c^{b'}) = 1$ or $\beta(s_{c'}^{b'}) = 1$ for $b' \in \mathsf{child}(b)$, if b is relabeling operation with for $c' \mapsto c$, or (3) $\beta(s_c^{b'}) = 1$ for $b' \in \mathsf{child}(b)$, if b edge introduction operation. Moreover, we define the evaluation of β over remaining variables in $E^* \cup A^*$ in the same manner as in the poof of

Theorem 10 via mapping the given admissible set to a satisfying assignment α for $\varphi_{\#\mathsf{adm}}$. Then, clearly β satisfies Formulas (26)–(29) by construction. Finally, β satisfies Formula 31 since $\beta(s_c^b) = 1$ in at initial leaf operation b due to the existence of an argument $a \in S' \setminus S$. But this leads to a contradiction to $\varphi_{\#\mathsf{pref}}[\alpha] \equiv 1$ since there can be no such assignment β. Consequently, for every satisfying assignment α to variables $E \cup A$ of $\varphi_{\#\mathsf{pref}}$, F admits exactly one preferred extension S. □

Theorem 16 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ k-expression. The reduction $\mathcal{R}_{\#\mathsf{pref}\to\#2\text{-QBF}}(F,\mathcal{X})$ constructs a QSAT instance ψ that linearly preserves the width, i.e.,* $\mathsf{dcw}\,\mathcal{G}_i^d(\mathsf{matrix}(\psi)) \in \mathcal{O}(k)$.

Proof. Building on top of Theorem 11, we highlight the changes due to Formulas (25)–(31).

First of all, we need $k' = 11k + 2$ colors to construct a k'-expression for $\varphi_{\#\mathsf{adm}}$. Then, the k'-expression for $\varphi^*_{\#\mathsf{adm}}$ requires the same number of colors, so that we can draw the incidence graph for $\varphi_{\#\mathsf{adm}}$ and $\varphi^*_{\#\mathsf{adm}}$ at the same time. Finally, to cover Formulas (26)–(31), we require a fresh color with all five versions as before, called *inequality* colors, denoted as $\{i_c^o, ic_c^o\} \cup \{ix_c\}$ for $o \in \{+,-\}$. Allowing different colors corresponding to variables in $\varphi_{\#\mathsf{adm}}$ and $\varphi^*_{\#\mathsf{adm}}$ had the advantage that clauses corresponding to Formulas (26) and (30) are also covered. This is the case since literals e_c^b and e_c^{*b} both can be colored differently, and hence the edges for corresponding clauses can be added.

Consequently, $2(11k+2)$ colors are needed due to $\varphi_{\#\mathsf{adm}}$ and $\varphi^*_{\#\mathsf{adm}}$, and additional $5k$ colors are required to cover Formulas (26)–(31). As a result, the total number of colors being $k^p = 27k + 4$ still increases only linearly in the construction of the k^p-expression $\mathcal{X}^p$ for $\varphi_{\#\mathsf{pref}}$. □

Semi-Stable Extensions

Theorem 17 ($\star$,Correctness)**.** *Let F be an AF and $\mathcal{X}$ be a k-expression of F. Then, the DDG reduction $\mathcal{R}_{\#\mathsf{semiSt}\to\#2\text{-QBF}}$ is correct, that is,* $\#\mathsf{semiSt}$ *on F coincides with* $\#2$-QBF *on* $\mathcal{R}_{\#\mathsf{semiSt}\to\#2\text{-QBF}}(F,\mathcal{X})$.

Proof. Once again, we begin by observing a one-one correspondence between admissible extension S in F and satisfying assignments α for $\varphi_{\#\mathsf{adm}}$ due to Theorem 10. Therefore, it suffices to prove that the Formulas (33)–(37) together with Formulas (5)–(8) allow range-maximal admissible (hence semi-stable) extensions.

"$\Longrightarrow$" Let S be a semi-stable extension in F. Then, we construct a unique satisfying assignment α for $\varphi_{\#\mathsf{semiSt}}$. We set, $\alpha(e_a) = 1$ for every $a \in S$ and $\alpha(e_a) = 0$ otherwise. Furthermore, for remaining variables in E and A, we map them exactly as in the proof of Theorem 10. Then, $\alpha \models \varphi_{\#\mathsf{adm}}$ since S is admissible. Finally, we map α to variables in D as in the proof of Theorem 7. That is, we set $\alpha(d_c^b) = 1$ for every color c and operation b of the k-expression, if and only if, (1) $\alpha(e_c^b) = 1$, if $b = c(a)$ is an initial k-graph with $c \in \mathsf{cols}(b)$; (2) $\alpha(d_c^{b'}) = 1$ for each $b' \in \mathsf{child}(b)$ s.t. $c \in \mathsf{cols}(b')$, if $b = \oplus$; (3) $\alpha(d_{c'}^{b'}) = 1$ and $\alpha(d_c^{b'}) = 1$, if $b = \rho_{c'\mapsto c}$ with $c \in \mathsf{cols}(b) \cap \mathsf{cols}(b')$; or (4) either $\alpha(d_c^{b'}) = 1$ or $\alpha(e_{c'}^b) = 1$, if $b = \eta_{c',c}$ with $c \in \mathsf{cols}(b)$. Then, α satisfies Formulas (5)–(8) by construction. Now, we prove that $\varphi_{\#\mathsf{semiSt}}[\alpha] \equiv 1$. Assume towards a contradiction that there exists an assignment β over variables in E^*, A^*, D^*S, such that $\varphi_{\#\mathsf{semiSt}}[\alpha][\beta] \equiv 1$. In particular, this implies that $\beta \models \varphi^*_{\#\mathsf{adm}}$ due to Formulas (32). Moreover, there is some color c such that $\beta(s_c^{rt}) = 1$. due to Formulas (37). However, this implies that either (1) there is some initial operation b such that $\beta(s_{c'}^b) = 1$ by Formulas (34)–(35), or (2) there is some edge introducing operation b such that $\beta(d_{c'}^{*b}) = 1$ but $\alpha(d_{c'}^b) = 0$ by Formulas (36), for some color (possibly) $c' = c$. But both cases implies that $\beta(d_c^{*b}) = 1$ and $\alpha(d_c^b) = 0$ must be true for at least one color c. We construct an admissible set S^* via the assignment β to variables in E^* by letting $S^* = \{a \mid a \in F, \beta(e_a^*) = 1\}$. Then S is indeed admissible since β satisfies $\varphi^*_{\#\mathsf{adm}}$

due to Formulas (32). Moreover, since $\varphi_{\#\mathsf{semiSt}}[\alpha][\beta] \equiv 1$, the two assignments together satisfy Formulas (37) and hence $S_R^+ \subseteq S_R^{*+} \subseteq A$ by definition of S^* and α. Furthermore, we have $S_R^+ \subsetneq S_R^{*+}$ due to Formula (33). However, this leads to a contradiction to S being semi-stable since S^* is also admissible and S_R^{*+} has more arguments than S_R^+. Consequently, there is no such assignment β satisfying Formulas (33)–(37). Finally, observe that there is no assignment α' over variables $E \cup A$ such that $\alpha' \neq \alpha$ and α' is a satisfying assignment for $\varphi_{\#\mathsf{semiSt}}$ if $\{a \mid a \in S\} = \{a \mid \alpha'(e_a) = 1\}$. This holds due to the same claim for $\varphi_{\#\mathsf{adm}}$, since there is bijection between admissible extensions and satisfying assignments for $\varphi_{\#\mathsf{adm}}$.

"$\Longleftarrow$". Let α be a satisfying assignment over $E \cup A$ such that $\varphi_{\#\mathsf{semiSt}}[\alpha] \equiv 1$. Then, we construct an extension $S = \{a \mid a \in A, \alpha(e_a) = 1\}$ and prove that S is a semi-stable in F. S is indeed admissible since $\alpha \models \varphi_{\#\mathsf{adm}}$. Now, assume towards a contradiction and let S' be another admissible set in F such that $S_R'^+ \supset S^+$. Then, we construct an assignment β using S' such that $\varphi_{\#\mathsf{semiSt}}[\alpha][\beta] \equiv 1$. For $a \in A$, we define $\beta(s_c^b) = 1$ if there is an argument $a \in A$ in the leaf operation of our k-expression with color c such that $a \in S_R'^+ \setminus S_R^+$, and we set $\beta(s_c^b) = 0$ otherwise. Furthermore, we for each non-leaf b, we set $\beta(s_c^b) = 1$ iff either (1) $\beta(s_c^{b'}) = 1$ for $b' \in \mathsf{child}(b)$, if b is disjoint union, or (2) either $\beta(s_c^{b'}) = 1$ or $\beta(s_{c'}^{b'}) = 1$ for $b' \in \mathsf{child}(b)$, if b is relabeling operation with for $c' \mapsto c$, or (3) either $\beta(s_c^{b'}) = 1$, or $\beta(d_c^{*b}) = 1$ and $\alpha(d_c^b) = 0$ for $b' \in \mathsf{child}(b)$, if b edge introduction operation. Moreover, we define the evaluation of β over remaining variables in $E^* \cup A^*$ (resp., D^*) in the same manner as in the poof of Theorem 10 (Theorem 7). Then, clearly β satisfies (5)–(8) and Formulas (33)–(36) by construction. Finally, β satisfies Formula 37 since $\beta(s_c^b) = 1$ in either some initial leaf operation or in some edge introduction operation b, due to the existence of an argument $a \in S'^+{}_R \setminus S_R^+$. But this leads to a contradiction to $\varphi_{\#\mathsf{semiSt}}[\alpha] \equiv 1$ since there is no such assignment β. Consequently, for every satisfying assignment α to variables $E \cup A$ of $\varphi_{\#\mathsf{semiSt}}$, F admits exactly one preferred extension S. □

Theorem 18 ($\star$,CW-Awareness)**.** *Let F be an AF and $\mathcal{X}$ a k-expression. The reduction $\mathcal{R}_{\#\mathsf{semiSt}\to\#2\text{-}\mathrm{QBF}}(F, \mathcal{X})$ constructs a QSAT instance ψ that linearly preserves the width, i.e.,* $\mathsf{dcw}(\mathcal{G}_i^d(\mathsf{matrix}(\psi))) \in \mathcal{O}(k)$.

Proof. The proof is analogous to the proof of Theorem 16 and builds on top of Theorem 11. We again highlight the changes due to Formulas (5)–(8) and Formulas (32)–(37). As before, we need $k' = 2(11k + 2)$ colors to construct a k'-expression for $\varphi_{\#\mathsf{adm}}$ and $\varphi^*_{\#\mathsf{adm}}$. To cover Formulas (33)–(37), we require fresh *inequality* colors $\{i_c^o, ic_c^o\} \cup \{ix_c\}$ for $o \in \{+, -\}$. However, now we need additional colors to cover *defeated* variables in Formulas (5)–(8), similar to the proof of Theorem 7. Consequently, we additionally require colors $\{d_c^o, dc_c^o\} \cup \{dx_c\}$ for $o \in \{+, -\}$ as in the case of stable semantics. Since we use different colors corresponding to variables in $\varphi_{\#\mathsf{adm}}$ and $\varphi^*_{\#\mathsf{adm}}$, as well as for variables in D and D^*, the Formulas (33) and (37) are all covered.

Consequently, $2(11k + 2)$ colors are needed due to $\varphi_{\#\mathsf{adm}}$ and $\varphi^*_{\#\mathsf{adm}}$, and additional $2(5k)$ are required to cover Formulas (33)–(37). As a result, the total number of colors being $k^s = 32k + 4$ still increases only linearly in the construction of the k^s-expression $\mathcal{X}^s$ for $\varphi_{\#\mathsf{semiSt}}$. □

Lower Bounds with Cliquewidth

Theorem 21 ($\star$,Admissible CW-LB)**.** *Unless ETH fails, we can not decide for an AF $F = (A, R)$ of directed clique-width w in time $2^{o(w)} \cdot poly(|A| + |R|)$ whether there exists an admissible extension of F containing argument a.*

Proof. We reduce from 3SAT by adapting a known reduction [23]. For every variable in φ we create arguments x and $\overline{x}$ and attacks $x \rightarrowtail \overline{x}$, $x \leftarrowtail \overline{x}$. Also, we create an argument sat and an argument c for every clause c, as well as attacks $c \rightarrowtail sat$. For every literal l in c, we add an attack $l \rightarrowtail c$. The construction ensures that if there is an admissible extension containing sat, all clauses need to be attacked, which works if we choose arguments that are consistent, satisfying φ.

Observe that the construction is such that F is indeed almost the directed incidence graph of φ. Indeed, for any given k-expression, we can double all colors, where the copy is for arguments $\overline{x}$ representing negated variables. Then, the attack / directed edge from $l \rightarrowtail c$ can be easily created whenever the corresponding directed edge is created. Also, $x \rightarrowtail \overline{x}$, $x \leftarrowtail \overline{x}$ can be easily covered in the leaves. Finally, $c \rightarrowtail sat$ can be covered in a leaf operation, where c is introduced, as we can easily just add an additional color for sat in every operation of the k-expression. Consequently, since the clique-width of F only linearly increases from w, the result holds. ☐

Proposition 28 (Preferred/Semi-Stable/Stage CW-LB)**.** *Under ETH we can not decide for an AF $F = (A, R)$ of directed clique-width w in time $2^{2^{o(w)}} \cdot poly(|A| + |R|)$ whether there exists a preferred (semi-stable/stage) extension of F that does not contain a (contains a).*

Proof (Idea). For preferred extensions, we modify a construction for skeptical acceptance of an argument [20], reducing from $\forall\exists$-QBF. For semi-stable and stage extensions, we modify a construction for credulous acceptance of an argument [23], which reduces from $\forall\exists$-QBF (and inverts). Both constructions are quite similar and slight modifications of the construction used in the proof of Theorem 21, but those do not significantly change the structure. Indeed, we can double the number of used colors (for auxiliary arguments per universal argument variable) and add a constant number of additional colors. The remainder works as in Theorem 21. ☐