# Reasoning in Argumentation Frameworks of Bounded Clique-Width [1]

Wolfgang DVOŘÁK, Stefan SZEIDER and Stefan WOLTRAN
*Institut für Informationssysteme,*
*Technische Universität Wien, A-1040 Vienna, Austria.*

**Abstract.** Most computational problems in the area of abstract argumentation are intractable, thus identifying tractable fragments and developing efficient algorithms for such fragments are important objectives towards practically efficient argumentation systems. One approach to tractability is to view abstract argumentation frameworks (AFs) as directed graphs and bound certain graph parameters. In particular, Dunne showed that many problems can be solved in linear time for AFs of bounded treewidth. In this paper we consider the graph-parameter clique-width, which is more general than treewidth. An additional advantage of clique-width over treewidth is that it applies well to directed graphs and takes the orientation of edges into account. We first give theoretical tractability results for AFs of bounded clique-width and then introduce dynamic-programming algorithms for credulous and skeptical reasoning.

**Keywords.** Computational Properties of Argumentation, Clique-Width, Fixed-Parameter Tractability

## 1. Introduction

Starting with the seminal work by Dung [14] the area of argumentation has evolved to one of the most active research branches within Artificial Intelligence (see, e.g., [2]). Dung's abstract argumentation frameworks, where arguments are seen as abstract entities which are just investigated with respect to their inter-relationship in terms of "attacks", are nowadays well understood and different semantics (i.e. the selection of sets of arguments which are jointly acceptable) have been proposed. However, most computational problems studied for such frameworks are intractable (see, e.g. [13,16]), while the importance of efficient algorithms for tractable fragments is evident. Symmetric [6] and bipartite argumentation frameworks [15] are such examples of fragments, where certain problems, although intractable in general, can efficiently be solved.

An interesting approach to dealing with intractable problems comes from parameterized complexity theory and is based on the fact, that many hard problems become polynomial-time tractable if some problem parameter is bounded by a fixed constant. In case the order of the polynomial bound on the runtime is independent of the parameter one speaks of *fixed-parameter tractability* (FPT). Since abstract argumentation frame-

[1] Dvořák's and Woltran's work was supported by the Vienna Science and Technology Fund (WWTF) under grant ICT08-028 and Szeider's work was supported by the European Research Council, grant reference 239962.

works are naturally represented as directed graphs, understanding the complexity of argumentation problems with respect to graph parameters is of high importance.

*Clique-width* is such a graph parameter that measures in a certain sense the structural complexity of a directed or undirected graph [9,10,12]. The parameter is defined via a graph construction process where only a limited number of vertex labels is available; vertices that share the same label at a certain point of the construction process must be treated uniformly in subsequent steps. Clique-width is related to the popular graph parameter *treewidth* [4]. Clique-width can be considered to be more general than treewidth since there are classes of graphs with constant clique-width but arbitrarily high treewidth (complete graphs, for instance). In contrast, graphs with bounded treewidth also have bounded clique-width [5,12].

Many NP-hard problems are tractable for graphs of bounded treewidth or clique-width. By means of a meta-theorem due to Courcelle, Makowsky, and Rotics [11] one can solve any graph problem that can be expressed in Monadic Second Order Logic with second-order quantification on vertex sets ($\text{MSO}_1$) in linear time for graphs of clique-width bounded by some fixed constant $k$. This result is similar to Courcelle's meta-theorem [7,8] for graphs of bounded treewidth.

First FPT results for argumentation used treewidth as parameter. While Dunne [15] was the first to give the necessary MSO characterizations, recent work [17] turned this theoretical tractability results (which follow from Courcelle's meta-theorem) into efficient dynamic programming algorithms. Moreover, [17] showed that for several other parameters (in particular, directed treewidth) bounding the parameter does not render the argumentation problems tractable.

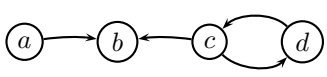
In this work, we focus on argumentation frameworks of bounded clique-width, which will lead us to a larger class of tractable argumentation frameworks. We emphasize that the MSO characterizations in [15] already yield tractability results for bounded clique-width (the involved MSO formulas are all from $\text{MSO}_1$). Thus, the main contribution in our paper is to introduce novel dynamic programming algorithms which put these theoretical results to work. For this purpose we establish particular succinct data structures (which we shall call (guarded) $k$-quadruples) characterizing the extensions of an argumentation framework. Due to the different nature of treewidth and clique-width these data structures differ significantly form those introduced in [17].

## 2. Argumentation Frameworks

In this section we introduce (abstract) argumentation frameworks [14], recall the preferred semantics for such frameworks, and highlight some known complexity results.

An *argumentation framework (AF)* is a pair $\mathcal{F} = (A, R)$ where $A$ is a set of arguments and $R \subseteq A \times A$ is the attack relation. We sometimes write $a \rightarrowtail b$ instead of $(a, b) \in R$, in case no ambiguity arises. Further, for $S \subseteq A$ and $a \in A$, we write $S \rightarrowtail a$ (resp. $a \rightarrowtail S$) iff there exists $b \in S$, such that $b \rightarrowtail a$ (resp. $a \rightarrowtail b$). An argument $a \in A$ is *defended* by a set $S \subseteq A$ iff for each $b \in A$, such that $b \rightarrowtail a$, also $S \rightarrowtail b$ holds.

*Example* 1. Let $\mathcal{F} = (A, R)$ with $A = \{a, b, c, d\}$ and $R = \{(a, b), (c, b), (c, d), (d, c)\}$. Thus, AFs can be represented as directed graphs. For instance, $\mathcal{F}$ looks like as follows:



**Definition 1.** *Let $\mathcal{F} = (A, R)$ be an AF. A set $S \subseteq A$ is* conflict-free (in $\mathcal{F}$) *if there are no $a, b \in S$, such that $(a, b) \in R$. A set $S \subseteq A$ is* admissible *for $\mathcal{F}$ if $S$ is conflict-free in $\mathcal{F}$ and each $a \in S$ is defended by $S$ in $\mathcal{F}$. $S$ is a* preferred extension *of $\mathcal{F}$ if $S$ is a maximal (wrt. subset inclusion) admissible set for $\mathcal{F}$. We denote the collection of all preferred extensions of $\mathcal{F}$ by $pref(\mathcal{F})$.*

For the AF $\mathcal{F}$ in Example 1, we get as admissible sets $\{\}$, $\{a\}$, $\{c\}$, $\{d\}$, $\{a, c\}$ and $\{a, d\}$. Consequently, $pref(\mathcal{F}) = \{\{a, c\}, \{a, d\}\}$.

Next, we recall the complexity of reasoning over preferred extensions. To this end, we define the decision problems of credulous and skeptical acceptance, which have as input an AF $\mathcal{F} = (A, R)$ and a set $S \subseteq A$ of arguments:

- CA: is there an extension $E \in pref(\mathcal{F})$, such that $S \subseteq E$;
- SA: does $S \subseteq E$ hold, for each $E \in pref(\mathcal{F})$?

It is known that CA is NP-complete, while SA is $\Pi_2^P$-complete (see [13,16]). The reason why CA is located on a lower level of the polynomial hierarchy compared to SA is the fact that it is sufficient to check whether $S$ is part of at least one admissible set for the given AF $\mathcal{F}$. Then $S \subseteq E$ also holds for a preferred extension $E$ of $\mathcal{F}$.

## 3. Clique-Width of Argumentation Frameworks

Let $k$ be a positive integer. A *$k$-graph* is a graph whose vertices are labeled by integers from $\{1, \ldots, k\} =: [k]$. The labeling of a graph $G = (V, E)$ is formally denoted by a function $\mathcal{L} : V \to [k]$. We consider an arbitrary graph as a $k$-graph with all vertices labeled by 1. We call the $k$-graph consisting of exactly one vertex $v$ (say, labeled by $i \in [k]$) an *initial $k$-graph* and denote it by $i(v)$.

Graphs can be constructed from initial $k$-graphs by means of repeated application of the following three operations.

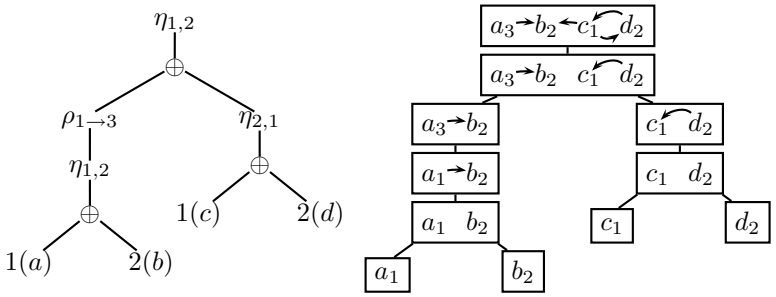
- *Disjoint union* (denoted by $\oplus$);
- *Relabeling*: changing all labels $i$ to $j$ (denoted by $\rho_{i \to j}$);
- *Edge insertion*: connecting all vertices labeled by $i$ with all vertices labeled by $j$ (denoted by $\eta_{i,j}$ or $\eta_{j,i}$); already existing edges are not doubled.[2]

A construction of a $k$-graph $G$ using the above operations can be represented by an algebraic term composed of $i(v)$, $\oplus$, $\rho_{i \to j}$, and $\eta_{i,j}$, ($i, j \in [k]$, and $v$ a vertex). Such a term is then called a *cwd-expression defining $G$*. A *$k$-expression* is a cwd-expression in which at most $k$ different labels occur. The set of all $k$-expressions is denoted by $CW_k$.

**Definition 2.** *The* clique-width *of a graph $G$,* cwd$(G)$*, is the smallest integer $k$ such that $G$ can be defined by a $k$-expression.*

For instance, trees have clique-width 3 and co-graphs have clique-width 2 (co-graphs are exactly given by the graphs which are $P_4$-free , i.e. whenever there is a path $(a, b, c, d)$ in the graph then $\{a, c\}$, $\{a, d\}$ or $\{b, d\}$ is also an edge of the graph).

[2] Some authors postulate that $i \neq j$ for the edge insertion $\eta_{i,j}$ to prohibit loops, but as AFs may have self-attacking arguments we do not.




(a) Parse tree of $\sigma$ (b) Labeled AFs for sub-expressions of $\sigma$

**Figure 1.** Example cwd-expression $\sigma = \eta_{1,2}(\rho_{1\to 3}(\eta_{1,2}(1(a) \oplus 2(b))) \oplus \eta_{2,1}(1(c) \oplus 2(d)))$

One can use $k$-expressions also to construct directed graphs (and thus argumentation frameworks), interpreting $\eta_{i,j}$ as the operation that inserts directed edges that are oriented from vertices labeled $i$ to vertices labeled $j$. Courcelle and Olariu [12] define the clique-width of a directed graph $G$ as the smallest integer $k$ such that $G$ can be constructed by a $k$-expression.

*Example* 2. The parse-tree of a cwd-expression $\sigma \in CW_3$ for the framework $\mathcal{F}$ from Example 1 is given in Figure 1(a). Figure 1(b) illustrates the labeled graphs associated to sub-expressions rooted in a node of the parse-tree. Here the index of a node denotes the current label, i.e. for $a$ such that $\mathcal{L}(a) = 3$ we write $a_3$. Actually one can show that $\text{cwd}(\mathcal{F}) = 2$, but for demonstrating our algorithms the above 3-expression is appropriate.

There are also classes of dense directed graphs which have low clique-width. As an example, we mention the class of transitive tournaments $\mathcal{T} = \{T_n : n \geq 1\}$ with $T_n = \big(\{a_1, \ldots, a_n\}, \{(a_i, a_j) : 1 \leq i < j \leq n\}\big)$. For an arbitrary $T_n$ we can give a cwd-expression that only uses two different labels. For illustration, we give such an expression defining the graph $T_5$ (see Figure 2):

$$\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(1(a_1) \oplus 2(a_2))) \oplus 2(a_3))) \oplus 2(a_4))) \oplus 2(a_5))$$

Similar 2-expressions exist for each tournament $T_n$, thus we obtain that the clique-width for each graph in $\mathcal{T}$ is bounded by 2.

*Modularity* is a further aspect of clique-width that makes it attractive in the context of abstract argumentation. Consider an AF $\mathcal{F} = (A, R)$. A subset $M \subseteq A$ is a *module* of $\mathcal{F}$ if any two arguments in $M$ are "indistinguishable from the outside," i.e., for any $x, x' \in M$ and $y \in A \setminus M$ we have $(x, y) \in R$ iff $(x', y) \in R$, and $(y, x) \in R$ iff $(y, x') \in R$. The AF $\mathcal{F}$ is called *prime* if all its modules are of size 1 or $|A|$. For instance, the AF $T_5$ of Figure 2 has a large module $\{a_2, \ldots, a_5\}$ and is therefore not prime. In AFs that model real-world situations one would expect to find large modules, for instance, formed by groups of people that share the same beliefs and opinions regarding the world outside the group (but possibly attack each other for some "internal" reason). One can handle AFs that contain large modules efficiently with clique-width based algorithms. Intuitively[3], from the cwd-expression of an AF $\mathcal{F}$ with a large module $M$ we can first find a cwd-expression for the subframework induced by $M$. Then we give all arguments in $M$

[3]This can be made precise: the clique-width of an AF equals the maximum clique-width of its prime subframeworks [12].

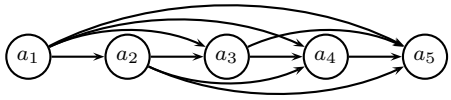



**Figure 2.** Tournament $T_5$

the same label and treat them for subsequent considerations as one single argument $a_M$. Consequently, for solving reasoning problems on $\mathcal{F}$ we can then strongly benefit from the "compression" of $M$ to $a_M$ when we use the clique-width based dynamic programming methods as described in Section 5.

Finally we summarize the properties that make clique-width an appealing parameter for abstract argumentation:

- There are both, sparse (e.g. tree-like AFs or AFs of bounded tree-width, see below) and dense AFs (e.g. transitive tournaments) that possess small clique-width.
- Clique-width incorporates the orientation of attacks.
- Clique-width offers an efficient handling for modular structures.

## 4. Fixed-Parameter Tractability

As we have already mentioned, clique-width is related to the popular graph parameter *treewidth* [4], therefore it is useful to review known results for the two parameters side by side. Clique-width can be considered to be more general than treewidth since there are graphs of constant clique-width but arbitrarily high treewidth (complete graphs, for instance), but graphs of bounded treewidth also have bounded clique-width [12,5]. An additional advantage of clique-width over treewidth is that it applies well to directed graphs and takes the orientation of edges into account.

By means of a meta-theorem due to Courcelle, Makowsky, and Rotics [11] one can solve any graph problem that can be expressed in Monadic Second Order Logic with second-order quantification on vertex sets ($\text{MSO}_1$) in linear time for graphs of clique-width bounded by some constant $k$. This result is similar to Courcelle's meta-theorem [7,8] which applies to a more general class of problems (problems expressible in Monadic Second Order Logic with second-order quantification on vertex sets and edge sets, $\text{MSO}_2$) on less general classes of graphs (graphs of bounded treewidth).

Treewidth and clique-width are both NP-hard to compute (as shown in [1] and [18], respectively). However, one can check in polynomial time whether the width of a given graph is bounded in terms of a fixed $k$. For treewidth this can be accomplished even exactly in linear time via Bodlaender's algorithm [3]. For clique-width, the known algorithms involve an additive approximation error that is bounded in terms of $k$: as shown by Oum and Seymour [20], there is a function $f$ such that one can find in polynomial time (of order independent of $k$) an $f(k)$-expression for an undirected graph of clique-width $k$. As shown by Kanté [19], a similar result holds also for directed graphs.

Dunne [15] already provided $\text{MSO}_2$ characterizations to show the fixed-parameter tractability of reasoning problems for argumentation frameworks of bounded treewidth. We observe that these $\text{MSO}_2$ characterizations do not make use of quantification over edge sets and so are in fact $\text{MSO}_1$ characterizations. Together with the meta-theorem for clique-width, we thus can immediately give the following result.

**Proposition 1.** *For AFs of clique-width bounded by a constant, the problems* CA *and* SA *are decidable in linear time.*

## 5. Algorithms

In this section we provide our dynamic-programming algorithms for credulous and skeptical acceptance. In fact, we start with credulous acceptance which relies on a simpler data structure (since we only have to characterize admissible sets), and then extend our ideas to skeptical acceptance. However, both algorithms follow the same basic principles by making use of a $k$-expression $\sigma$ defining an argumentation framework $\mathcal{F}$ in the following way: we assign certain objects (e.g. for CA we use $k$-quadruples as defined in Definition 3 below) to each subexpression of $\sigma$. We manipulate these objects in a bottom-up traversal of the parse tree of the $k$-expression such that the objects in the root of the parse tree then provide the necessary information to decide the problem under consideration. The size of these objects is bounded in terms of $k$ (and independent of the size of $\mathcal{F}$) and the number of such objects required is linear in the size of $\mathcal{F}$. Most importantly, we will show that these objects can also be efficiently computed for bounded $k$. Thus, we obtain the desired linear running time for AFs of bounded clique-width.

In what follows, we consider (unless stated otherwise) an AF $\mathcal{F} = (A, R)$ as a labeled directed graph, where the labeling is given by $\mathcal{L} : A \to [k]$ with appropriate $k$.

### *5.1. Credulous Acceptance*

**Definition 3.** *A tuple $Q = (I, A, O, D)$ with $I, A, O, D \subseteq [k]$ is called a* $k$-quadruple*, and we refer to its parts using $Q_{in} = I$, $Q_{att} = A$, $Q_{out} = O$ and $Q_{def} = D$. The set of all $k$-quadruples is given by $\mathcal{Q}_k$.*

The "semantics" of a $k$-quadruple $Q$ with respect to a given AF $\mathcal{F}$ is given as follows.

**Definition 4.** *Let $Q \in \mathcal{Q}_k$ and $\mathcal{F} = (A, R)$ be an AF labeled by $\mathcal{L} : A \to [k]$. An* $\mathcal{F}$-extension *of $Q$ is a conflict-free set $E \subseteq A$ satisfying:*

$$
\begin{aligned}
Q_{in} &= \{\mathcal{L}(a) : a \in E\} \\
Q_{att} &= \{\mathcal{L}(a) : a \in A \setminus E, a \rightarrowtail E, E \not\rightarrowtail a\} \\
Q_{out} &= \{\mathcal{L}(a) : a \in A \setminus E, E \not\rightarrowtail a, b \not\rightarrowtail E \text{ or } E \rightarrowtail b \text{ for all } b \text{ with } \mathcal{L}(b){=}\mathcal{L}(a)\} \\
Q_{def} &= \{\mathcal{L}(a) : a \in A \setminus E, E \rightarrowtail a, E \rightarrowtail b \text{ for all } b \text{ with } \mathcal{L}(b){=}\mathcal{L}(a)\}
\end{aligned}
$$

*The set of all $\mathcal{F}$-extensions of $Q$ is denoted by $\mathcal{E}_{\mathcal{F}}(Q)$. If $\mathcal{E}_{\mathcal{F}}(Q) \neq \emptyset$ we call the $k$-quadruple $Q$* valid *for $\mathcal{F}$.*

Informally speaking, for a given AF $\mathcal{F}$, a $k$-quadruple $Q$ characterizes sets $E$ such that for each $l \in Q_{in}$, at least one argument with $\mathcal{L}(a) = l$ is contained in $E$. Moreover, for arguments not contained in $E$, the sets $Q_{att}, Q_{out}, Q_{def}$ provide some further information about the relationship between arguments with respect to their labels. Let us mention here that for a valid $k$-quadruple $Q$, a label $l \in [k]$ may be contained in $Q_{in}$ and also in one of $Q_{att}, Q_{out}, Q_{def}$; however, $Q_{att}$, $Q_{out}$ and $Q_{def}$ are pairwise disjoint. It is important to observe that for each $k$ there is only a finite number of $k$-quadruples. With this finite number of $k$-quadruples we are able to represent an unbounded number of different sets $E$.

To further illustrate the idea behind $k$-quadruples, consider the AF $\mathcal{F}$ from Example 1 with labels $\mathcal{L}(a) = 3$, $\mathcal{L}(b) = \mathcal{L}(d) = 2$ and $\mathcal{L}(c) = 1$ as depicted in the root of

the tree in Figure 1(b). Consider $Q = (\{1,3\}, \emptyset, \emptyset, \{2\})$ and let us construct a set $E$ such that $E \in \mathcal{E}_{\mathcal{F}}(Q)$. We have only a single argument with label 1 and resp. with label 3, thus $E = \{a, c\}$. Moreover, both arguments with label 2, $b$ and $d$, satisfy the condition for $Q_{def}$, i.e. we have $E \rightarrowtail b$ and $E \rightarrowtail d$. Hence, $E \in \mathcal{E}_{\mathcal{F}}(Q)$ holds. As a second example, consider $Q' = (\{2\}, \{1,3\}, \{2\}, \emptyset)$. Since there are two arguments with label 2, we have three candidates for being an $\mathcal{F}$-extension $E'$ of $Q'$, namely $\{b\}$, $\{d\}$ and $\{b, d\}$ (all of them are conflict-free in $\mathcal{F}$). However, since $1 \in Q'_{att}$, $c \rightarrowtail E'$ and $E' \not\rightarrowtail c$ has to hold. Thus, $d$ cannot be contained in an $\mathcal{F}$-extension of $Q'$. Checking the remaining properties, one can show that $\mathcal{E}_{\mathcal{F}}(Q') = \{\{b\}\}$. Finally, consider $Q'' = (\{1,2\}, \{2,3\}, \emptyset, \emptyset)$ and suppose an $E'' \in \mathcal{E}_{\mathcal{F}}(Q'')$. Then $c \in E''$, since $c$ is the only argument with label 1. However, also $b$ or $d$ has to be contained in $E''$ since $2 \in Q''_{in}$. But then, $E''$ is not conflict-free in $\mathcal{F}$. Thus $\mathcal{E}_{\mathcal{F}}(Q'') = \emptyset$, i.e. $Q''$ is not valid for $\mathcal{F}$.

The following definition assigns $k$-quadruples (and certain relations between them) to the nodes of the parse-tree of a given $k$-expression. In what follows, we denote the AF defined by a $k$-expression $\sigma$ as $\mathcal{F}_\sigma$.

**Definition 5.** *A* full $k$-quadruple assignment *for a $k$-expression $\sigma$ is a function that maps each subexpression $\theta$ of $\sigma$ to a relation $L_\theta$ over $k$-quadruples in the following way:*

- *$L_\theta(Q, R, S)$ iff $\theta = \theta_1 \oplus \theta_2$, $Q$, $R$, $S$ are valid $k$-quadruples for $\mathcal{F}_\theta$, $\mathcal{F}_{\theta_1}$, and resp. $\mathcal{F}_{\theta_2}$, and for each $E_1 \in \mathcal{E}_{\mathcal{F}_{\theta_1}}(R)$, $E_2 \in \mathcal{E}_{\mathcal{F}_{\theta_2}}(S)$, also $E_1 \cup E_2 \in \mathcal{E}_{\mathcal{F}_\theta}(Q)$;*
- *$L_\theta(Q, Q')$ iff either $\theta = \rho_{i \to j}(\theta')$ or $\theta = \eta_{i,j}(\theta')$, $Q$ and $Q'$ are valid $k$-quadruples for $\mathcal{F}_\theta$ and resp. $\mathcal{F}_{\theta'}$, and for each $E \in \mathcal{E}_{\mathcal{F}'_\theta}(Q')$, also $E \in \mathcal{E}_{\mathcal{F}_\theta}(Q)$;*
- *$L_{l(v)}(Q)$ iff $Q$ is a valid $k$-quadruple of $\mathcal{F}_{l(v)}$.*

The following theorem reveals the benefits of full $k$-quadruple assignments for our algorithmic purposes. Indeed, the desired dynamic programming algorithm for $\mathsf{CA}$ is used within the proof of that result. We bound the running time of our algorithms in terms of the length of the given cwd-expression (which is linear in the size of the given AF).

**Theorem 1.** *Let $k$ be a constant.*

1. *Given a $k$-expression $\sigma$ for an AF $\mathcal{F}$, we can compute the full $k$-quadruple assignment for $\sigma$ in linear time.*
2. *Given the full $k$-quadruple assignment for $\sigma$ and a set $S$ of arguments, we can decide in linear time whether some preferred extension of $\mathcal{F}$ contains $S$.*

In the remainder of the subsection, we sketch a proof of this result and illustrate the central concepts using Example 1.

For the first part of the theorem we have to establish valid $k$-quadruples for all subexpressions $\theta$ of $\sigma$ and the respective relations $L_\theta$ without an explicit computation of $\mathcal{F}$-extensions. Instead, we use a bottom-up computation along the parse tree of $\sigma$. To this end, we recursively define a function $F_{\mathsf{CA}}$ which associates to each subexpression of $\sigma$ the set of $k$-quadruples which are valid for the respective subframework.

**Definition 6.** *The function $F_{\mathsf{CA}} : CW_k \to 2^{\mathcal{Q}_k}$ is recursively defined along the structure of $k$-expressions as follows.*

- $F_{\mathsf{CA}}(i(v)) = \{(\{i\}, \emptyset, \emptyset, \emptyset), (\emptyset, \emptyset, \{i\}, \emptyset)\}$
- $F_{\mathsf{CA}}(\sigma_1 \oplus \sigma_2) = \{Q \oplus^{\mathsf{CA}} Q' : Q \in F_{\mathsf{CA}}(\sigma_1), Q' \in F_{\mathsf{CA}}(\sigma_2)\}$ *where*

$$Q \oplus^{\mathsf{CA}} Q' = (Q_{in} \cup Q'_{in}, Q_{att} \cup Q'_{att}, (Q_{out} \cup Q'_{out}) \setminus (Q_{att} \cup Q'_{att}), Q_{def} \cap Q'_{def})$$

- $F_{\mathsf{CA}}(\rho_{i\to j}(\sigma)) = \{\rho^{\mathsf{CA}}_{i\to j}(Q') : Q' \in F_{\mathsf{CA}}(\sigma)\}$ *where* $\rho^{\mathsf{CA}}_{i\to j}(Q') = Q$ *holds iff the following conditions are jointly satisfied:*
  - $Q_{in} = \begin{cases} Q'_{in} \setminus \{i\} \cup \{j\} & \text{if } i \in Q'_{in} \\ Q'_{in} & \text{otherwise} \end{cases}$
  - $Q_{att} = \begin{cases} Q'_{att} \setminus \{i\} \cup \{j\} & \text{if } i \in Q'_{att} \\ Q'_{att} & \text{otherwise} \end{cases}$
  - $Q_{out} = \begin{cases} Q'_{out} \setminus \{i\} \cup \{j\} & \text{if } i \in Q'_{out} \text{ and } j \notin Q'_{att} \\ Q'_{out} \setminus \{i,j\} & \text{if } i \in Q'_{att} \\ Q'_{out} \setminus \{i\} & \text{otherwise} \end{cases}$
  - $Q_{def} = \begin{cases} Q'_{def} \setminus \{i\} \cup \{j\} & \text{if } i \in Q'_{def} \text{ and } j \notin Q'_{att} \cup Q'_{out} \\ Q'_{def} \setminus \{j\} & \text{otherwise} \end{cases}$
- $F_{\mathsf{CA}}(\eta_{i,j}(\sigma)) = \{\eta^{\mathsf{CA}}_{i,j}(Q) : Q \in F_{\mathsf{CA}}(\sigma), \{i,j\} \not\subseteq Q_{in}\}$ *where*
  $\eta^{\mathsf{CA}}_{i,j}(Q) = \begin{cases} (Q_{in}, Q_{att} \setminus \{j\}, Q_{out} \setminus \{j\}, Q_{def} \cup \{j\}) & \text{if } i \in Q_{in} \\ (Q_{in}, Q_{att} \cup \{i\}, Q_{out} \setminus \{i\}, Q_{def} \setminus \{i\}) & \text{if } j \in Q_{in},\ i \in Q_{out} \\ Q & \text{otherwise} \end{cases}$

By definition, $F_{\mathsf{CA}}$ provides the necessary relations for the desired full $k$-quadruple assignment. Figure 3 illustrates the function $F_{\mathsf{CA}}$ for the cwd-expression used in Example 2 to define the AF $\mathcal{F}$. Each table in Figure 3 shows the valid $k$-quadruples for the subframeworks of $\mathcal{F}$ (the tree of tables mirrors the tree of subframeworks as given in Figure 1(b)). For a better understanding, we provide in Figure 3 also the extension[4] $\mathcal{E}$ of the $k$-quadruples; however, those extensions are not required for computing $F_{\mathsf{CA}}$.

Let us have a look at a few aspects of the computation of $F_{\mathsf{CA}}$. We start at the leaf nodes. For instance, in the leaf of the left branch, we consider the subframework defined by the expression $\sigma_1 = 1(a)$, i.e. the framework $\mathcal{F}_1 = (\{a\}, \emptyset)$ with $a$ labeled by 1. By definition of $F_{\mathsf{CA}}$, two $k$-quadruples are assigned to $\sigma_1$, viz. $Q_1 = (\{1\}, \emptyset, \emptyset, \emptyset)$ and $Q'_1 = (\emptyset, \emptyset, \{1\}, \emptyset)$ representing the two conflict-free sets of $\mathcal{F}_1$, $\{a\}$ and $\emptyset$. Similar $k$-quadruples (see the rhs sibling) are assigned to the framework $(\{b\}, \emptyset)$ defined by $\sigma_2 = 2(b)$. Then, $F_{\mathsf{CA}}(\sigma_1 \oplus \sigma_2)$ combines these $k$-quadruples according to Definition 6. In the next step we have to deal with an edge-insertion, i.e. we compute $F_{\mathsf{CA}}(\eta_{1,2}(\sigma_1 \oplus \sigma_2))$; observe that the $k$-quadruple $Q = (\{1,2\}, \emptyset, \emptyset, \emptyset)$ thus drops out by definition, since $\{1,2\} \subseteq Q_{in}$. Also observe that the status of the arguments which have been "out" changes accordingly to the direction of the introduced edge. Due to space restrictions, we cannot give a discussion about the entire bottom-up computation of $F_{\mathsf{CA}}$ here, but we mention that each $k$-quadruple as computed by $F_{\mathsf{CA}}$ in Figure 3 is indeed valid for the respective subframework. As well, all the relations from Definition 5 are established by computing $F_{\mathsf{CA}}$. The general result is as follows.

**Lemma 1.** *Let $\sigma$ be a $k$-expression defining an AF $\mathcal{F}$. Then, $F_{\mathsf{CA}}(\sigma)$ coincides with the set of valid $k$-quadruples for $\mathcal{F}$. Further, the valid $k$-quadruples together with the operators $\oplus^{\mathsf{CA}}$, $\rho^{\mathsf{CA}}_{i\to j}$,$\eta^{\mathsf{CA}}_{i,j}$ give the full $k$-quadruple assignment for $\sigma$.*

[4] In this example, there is always exactly one such extension for the depicted valid $k$-quadruples; hence $\emptyset$ refers to the empty extension and not to the empty set of extensions.

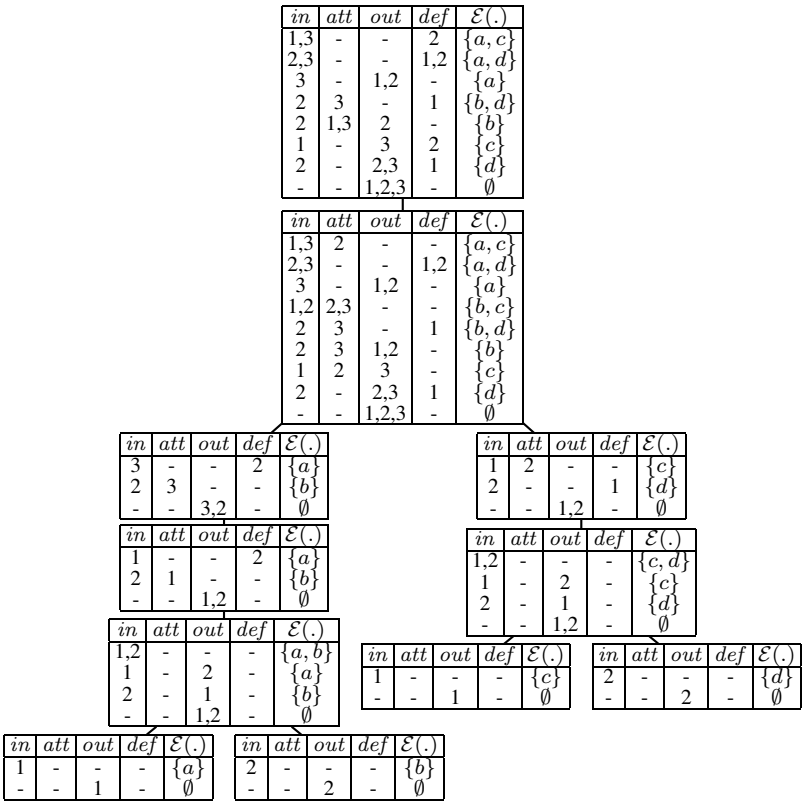



**Figure 3.** The function $F_{\mathsf{CA}}$ for the k-expression in Example 2

The time to compute $F_{\mathsf{CA}}(\sigma)$ depends (heavily) on $k$ but is linear in the size of $\sigma$ (and thus in the size of $\mathcal{F}_\sigma$). This concludes the proof sketch for (1).

To prove the second part of Theorem 1, we require an appropriate connection between admissible sets and $k$-quadruples.

**Lemma 2.** *Let $\mathcal{F}$ be an AF and $E$ be conflict-free in $\mathcal{F}$. Then there is a unique valid $k$-quadruple $Q$ with $E \in \mathcal{E}_\mathcal{F}(Q)$. Further, $E$ is an admissible extension of $\mathcal{F}$ iff $Q_{att} = \emptyset$.*

Given the full $k$-quadruple assignment for $\sigma$, we are now able to efficiently decide whether an argument set $S$ is contained in at least one admissible extension of $\mathcal{F}$: For each node of the parse-tree, i.e. each subexpression $\sigma'$, we mark the valid $k$-quadruples that represent at least one admissible extension $E$ of the subframework $\mathcal{F}_{\sigma'} = (A', R')$ such that $S \cap A' \subseteq E$. This can be done as follows: First, for each cwd-expression $i(v)$, $(\{i\}, \emptyset, \emptyset, \emptyset)$ is marked in case $v \in S$; otherwise, both $(\{i\}, \emptyset, \emptyset, \emptyset)$ and $(\emptyset, \emptyset, \{i\}, \emptyset)$ are marked. Hence, quadruples without a mark indicate that some element from $S$ is missing. In the $\eta_{i,j}$ and $\rho_{i \to j}$ nodes of the parse tree, a quadruple is marked if it is in relation with a marked quadruple in the child node. For a $\oplus$-node (i.e. for a subexpression $\theta$ with $\oplus$ as main connective), we mark the quadruple $Q$ if $L_\theta(Q, Q_1, Q_2)$ holds for some marked quadruples $Q_1$, $Q_2$. If we have a marked quadruple $Q$ in the root and this quadruple represents an admissible extension, i.e. $Q_{att} = \emptyset$ holds, then $S$ is credulously accepted.

This marking algorithm is obviously running in linear time w.r.t. to the size of the full $k$-quadruple assignment of $\mathcal{F}$. This shows the second part of the theorem.

### *5.2. Skeptical Reasoning*

For skeptical acceptance we augment each $k$-quadruple by a so-called guard which stores quadruples representing larger (wrt. $\subseteq$) extensions. This will allow us to characterize not only admissible but also preferred extensions.

**Definition 7.** *A* guarded $k$-quadruple *is a pair $(Q, \Gamma)$ where $Q \in \mathcal{Q}_k$ and $\Gamma \subseteq \mathcal{Q}_k$ is the guard for $Q$. The set of all guarded $k$-quadruples is given by $G\mathcal{Q}_k$.*

**Definition 8.** *Let $(Q, \Gamma) \in G\mathcal{Q}_k$ and $\mathcal{F} = (A, R)$ be an AF. An* $\mathcal{F}$-extension *of $(Q, \Gamma)$ (in $\mathcal{F}$), is a conflict-free set $E \subseteq A$ in $\mathcal{F}$ satisfying: (1) $E \in \mathcal{E}_{\mathcal{F}}(Q)$; (2) for each conflict-free set $E'$ of $\mathcal{F}$ with $E \subset E'$, there is a $Q' \in \Gamma$ such that $E' \in \mathcal{E}_{\mathcal{F}}(Q')$; and (3) for each $Q' \in \Gamma$ there exists an $E'$ with $E \subset E'$, such that $E' \in \mathcal{E}_{\mathcal{F}}(Q')$.*

*The set of all $\mathcal{F}$-extensions of $(Q, \Gamma)$ (in $\mathcal{F}$) is denoted by $\mathcal{E}_{\mathcal{F}}(Q, \Gamma)$. If $\mathcal{E}_{\mathcal{F}}(Q, \Gamma) \neq \emptyset$ we call $(Q, \Gamma)$* valid *for $\mathcal{F}$.*

Replacing in Definition 5 $k$-quadruples by guarded $k$-quadruples and $\mathcal{E}(Q)$ by $\mathcal{E}(Q, \Gamma)$ gives us the concept of a full guarded $k$-quadruple assignment.

**Theorem 2.** *Let $k$ be a constant.*

1. *Given a $k$-expression $\sigma$ for an AF $\mathcal{F}$, we can compute the full guarded $k$-quadruple assignment for $\sigma$ in linear time.*
2. *Given the full guarded $k$-quadruple assignment for $\sigma$ and a set $S$ of arguments, we can decide in linear time whether all preferred extensions of $\mathcal{F}$ contain $S$.*

As before, we now provide a function, which recursively establishes the full guarded $k$-quadruple assignment without an explicit computation of extensions (for $\Gamma, \Gamma' \subseteq \mathcal{Q}_k$, we below use the operator $\Gamma \oplus^{\mathsf{SA}} \Gamma' = \{Q \oplus^{\mathsf{CA}} Q' : Q \in \Gamma, Q' \in \Gamma'\}$).

**Definition 9.** *The function $F_{\mathsf{SA}} : CW_k \to 2^{G\mathcal{Q}_k}$ is recursively defined as follows.*

- $F_{\mathsf{SA}}(i(v)) = \big\{\big(\{i\}, \emptyset, \emptyset, \emptyset\big),\ \big(\emptyset, \emptyset, \{i\}, \emptyset\big), \{(\{i\}, \emptyset, \emptyset, \emptyset)\}\big\}$
- $F_{\mathsf{SA}}(\sigma_1 \oplus \sigma_2) = \big\{\big(Q_1 \oplus^{\mathsf{CA}} Q_2, (\Gamma_1 \oplus^{\mathsf{SA}} \Gamma_2) \cup (\{Q_1\} \oplus^{\mathsf{SA}} \Gamma_2) \cup (\Gamma_1 \oplus^{\mathsf{SA}} \{Q_2\})\big) : (Q_1, \Gamma_1) \in F_{\mathsf{SA}}(\sigma_1), (Q_2, \Gamma_2) \in F_{\mathsf{SA}}(\sigma_2)\big\}$
- $F_{\mathsf{SA}}(\rho_{i \to j}(\sigma)) = \big\{\big(\rho^{\mathsf{CA}}_{i \to j}(Q), \{\rho^{\mathsf{CA}}_{i \to j}(Q') : Q' \in \Gamma\}\big) : (Q, \Gamma) \in F_{\mathsf{SA}}(\sigma)\big\}$
- $F_{\mathsf{SA}}(\eta_{i,j}(\sigma)) = \big\{\big(\eta^{\mathsf{CA}}_{i,j}(Q), \{\eta^{\mathsf{CA}}_{i,j}(Q') : Q' \in \Gamma, \{i, j\} \not\subseteq Q'_{in}\}\big) : (Q, \Gamma) \in F_{\mathsf{SA}}(\sigma), \{i, j\} \not\subseteq Q_{in}\big\}$

Roughly speaking, for $(Q, \Gamma) \in G\mathcal{Q}_k$ we apply here the already defined function $F_{\mathsf{CA}}$ not only to $Q$ but also to each element in $\Gamma$. It can be shown that $F_{\mathsf{SA}}(\sigma)$ coincides with the set of valid guarded $k$-quadruples for $\mathcal{F}_\sigma$. Further the valid guarded k-quadruples together with the operators $\oplus^{\mathsf{SA}}$, $\rho^{\mathsf{SA}}_{i \to j}$, $\eta^{\mathsf{SA}}_{i,j}$ give the full guarded $k$-quadruple assignment for $\sigma$. We note that due to the use of guards, computing $F_{\mathsf{SA}}(\sigma)$ is more involved compared to $F_{\mathsf{CA}}(\sigma)$, for a given $k$-expression $\sigma$. However, the size of the tables for each node in the parse-tree of $\sigma$ still remains bound by $k$ and is independent from the actual size of $\sigma$. This guarantees a linear-running time with respect to the size of AFs defined by $\sigma$ also for $F_{\mathsf{SA}}(\sigma)$ as long as the AFs have their clique-width bounded by $k$.

To show (2) we give a certain link between preferred sets and guarded $k$-quadruples.

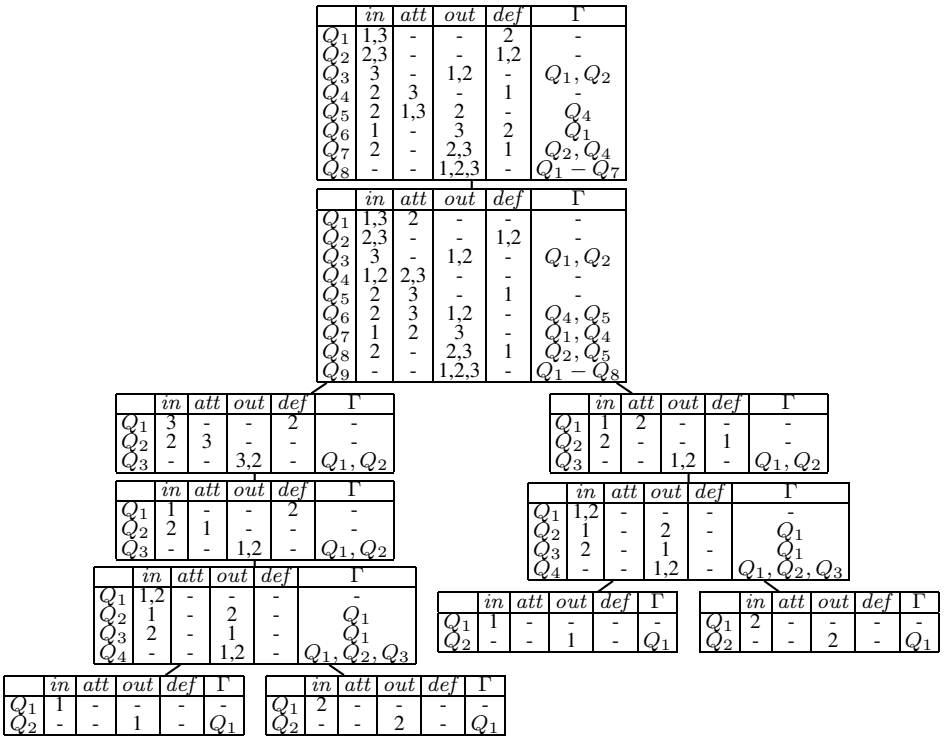



**Figure 4.** The function $F_{\mathsf{SA}}$ for Example 2

**Lemma 3.** *Let $\mathcal{F}$ be an AF and let $E$ be conflict-free in $\mathcal{F}$. Then there exists a unique valid guarded $k$-quadruple $(Q, \Gamma)$ such that $E \in \mathcal{E}_{\mathcal{F}}(Q, \Gamma)$. Moreover, $E$ is a preferred extension of $\mathcal{F}$ iff $Q_{att} = \emptyset$ and there is no $Q' \in \Gamma$ such that $Q'_{att} = \emptyset$.*

Figure 4 illustrates the function $F_{\mathsf{SA}}$ for our running example. Compared to Figure 3 we now give also the guard for each $k$-quadruple. Due to space restrictions, a detailed discussion of this example has to be omitted. We just note that there are four valid guarded $k$-quadruples $(Q, \Gamma)$ in the root which match the condition that there is no $Q' \in \Gamma$ such that $Q'_{att} = \emptyset$, namely $G_1 = ((\{1,3\}, \emptyset, \emptyset, \{2\}), \emptyset)$, $G_2 = ((\{2,3\}, \emptyset, \emptyset, \{1,2\}), \emptyset)$, $G_4 = ((\{2\}, \{3\}, \emptyset, \{1\}), \emptyset)$ and $G_5 = ((\{2\}, \{1,3\}, \{2\}, \emptyset), \{(\{2\}, \{3\}, \emptyset, \{1\})\})$, with their extensions $\mathcal{E}_{\mathcal{F}}(G_1) = \{\{a, c\}\}$, $\mathcal{E}_{\mathcal{F}}(G_2) = \{\{a, d\}\}$, $\mathcal{E}_{\mathcal{F}}(G_4) = \{\{b, d\}\}$ and $\mathcal{E}_{\mathcal{F}}(G_5) = \{\{b\}\}$. $G_1$, $G_2$ and $G_4$ thus characterize maximal conflict-free sets of $\mathcal{F}$, but $G_4$ is identified to be not preferred since $(G_4)_{att} = \{3\} \neq \emptyset$. We also have selected $G_5$, since there is no superset of $\{b\}$ admissible in $\mathcal{F}$; however, we also have $(G_5)_{att} = \{1, 3\} \neq \emptyset$. Thus, $G_1$ and $G_2$ are the only ones fulfilling all necessary conditions for characterizing preferred extensions of $\mathcal{F}$, which indeed are $\{a, c\}$ and $\{a, d\}$.

The algorithm for skeptical acceptance is similar to the one for CA discussed above. The only pairs marked in leafs $i(v)$ are $((\emptyset, \emptyset, \{i\}, \emptyset), \{(\{i\}, \emptyset, \emptyset, \emptyset)\})$ for $v \in S$. In the other nodes, a guarded $k$-quadruple is marked if it is in relation with at least one marked pair in its child(ren). Now, there is a marked pair for the root representing a preferred extension (cf. Lemma 3) exactly if $S$ is *not* skeptically accepted, since in this case we have found a preferred extension where at least one argument from $S$ was left out.

## 6. Conclusion

In this paper, we turned some theoretical tractability results (which implicitly follow from previous work [15]) for argumentation frameworks of bounded clique-width into efficient algorithms. These algorithms are applicable to arbitrary frameworks, whenever a defining $k$-expression is given, but the runtime heavily depends on $k$, rather than the size of the AF. Thus the algorithms are expected to run efficiently in particular for small $k$.

We restricted ourselves here to the problem of acceptance with respect to the preferred semantics, which relies on maximal admissible sets. However, admissibility and maximality are prototypical properties common in many other argumentation semantics. Hence, we expect that the methods developed here can also be extended to other semantics and reasoning tasks, which is left for future work.

## References


[1] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a $k$-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, 1987.

[2] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[3] H. L. Bodlaender. On linear time minor tests with depth-first search. *J. Algorithms*, 14(1):1–23, 1993.

[4] H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–21, 1993.

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

[6] S. Coste-Marquis, C. Devred, and P. Marquis. Symmetric argumentation frameworks. In *Proc. ECSQARU'05*, volume 3571 of *LNCS*, pages 317–328. Springer, 2005.

[7] B. Courcelle. Recognizability and second-order definability for sets of finite graphs. Technical Report I-8634, Université de Bordeaux, 1987.

[8] B. Courcelle. Graph rewriting: an algebraic and logic approach. In *Handbook of Theoretical Computer Science, Vol. B*, pages 193–242. Elsevier Science Publishers 1990.

[9] B. Courcelle, J. Engelfriet, and G. Rozenberg. Context-free handle-rewriting hypergraph grammars. In *Proc. Graph Grammars 1990*, volume 532 of *LNCS*, pages 253–268. Springer, 1991.

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

[11] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

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

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

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

[15] P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15):701–729, 2007.

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

[17] W. Dvořák, R. Pichler, and S. Woltran. Towards fixed-parameter tractable algorithms for argumentation. In *Proc. KR'10*, pages 112-122. AAAI Press, 2010

[18] M. R. Fellows, F. A. Rosamond, U. Rotics, and S. Szeider. Clique-width is NP-complete. *SIAM J. Discrete Math.*, 23(2):909–939, 2009.

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

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