Investigating Subclasses of Abstract Dialectical Frameworks

Martin DILLER [a], Atefeh KESHAVARZI ZAFARGHANDI [b], Thomas LINSBICHLER [a], and Stefan WOLTRAN [a 1]

[a] *Institute of Logic and Computation, TU Wien, Austria*

[b] *Institute of Artificial Intelligence, University of Groningen, The Netherlands*

Abstract. Abstract dialectical frameworks (ADFs) are generalizations of Dung argumentation frameworks where arbitrary relationships among arguments can be formalized. This additional expressibility comes with the price of higher computational complexity, thus an understanding of potentially easier subclasses is essential. Compared to Dung argumentation frameworks, where several subclasses such as acyclic and symmetric frameworks are well understood, there has been no in-depth analysis for ADFs in such direction yet (with the notable exception of bipolar ADFs). In this work, we introduce certain subclasses of ADFs and investigate their properties. In particular, we show that for acyclic ADFs, the different semantics coincide. On the other hand, we show that the concept of symmetry is less powerful for ADFs and further restrictions are required to achieve results that are similar to the known ones for Dung's frameworks. We also provide experiments to analyse the performance of solvers when applied to particular subclasses of ADFs.

Keywords. abstract argumentation, abstract dialectical frameworks, acyclic graphs, expressiveness, performance analysis

1. Introduction

Abstract dialectical frameworks (ADFs) are generalizations of Dung argumentation frameworks (AFs) [2] where arbitrary relationships among arguments can be formalized via propositional formulas which are attached to the arguments [3,4]. This allows to formalize notions of support, collective attacks, and even more complicated relations. Due to their flexibility in fomalizing relations between arguments, ADFs have recently been used in several applications [5,6,7,8]. However, this additional expressibility comes with the price of higher computational complexity [9]. Specifically, reasoning in ADFs spans the first three (rather than the first two, as for AFs) levels of the polynomial hierarchy.

It is thus a natural question to investigate subclasses of ADFs. Compared to Dung argumentation frameworks, where several subclasses have been thoroughly studied (see e.g., [10,11,12]), there has not been a systematic investigation of subclasses of ADFs yet, with the exception of bipolar ADFs [3] where the links between arguments are restricted to have either supporting or attacking nature. In particular, there have been no studies yet about under which structural restrictions on ADFs different semantics coincide.

[1] Author names are sorted in alphabetical order. This paper is based on the second author's master thesis [1].

In this work, we aim to define several subclasses of ADFs and investigate their properties in terms of semantics. As a first class, we consider acyclic ADFs (i.e., the link-structure forms an acyclic graph) and show that - analogous to well-founded AFs - the main semantics coincide for this class. We further investigate the concept of symmetric ADFs. In contrast to the case of AFs, we will see that properties as coherence and relative-groundedness do not carry over and require further restrictions which leads to the class of acyclic support symmetric ADFs (ASSADFs). Following the work of Dunne et al. [13] we investigate the expressiveness of our ASSADF subclass in terms of signatures, i.e. the set of possible outcomes which can be achieved by ASSADFs under the different semantics. We thus complement here results which have been obtained for general [14,15] and bipolar ADFs [16] and also compare ASSADFs with AFs in terms of expressibility. Finally, we provide a preliminary experimental analysis which analyses to which extent ADF-solvers benefit from instances that are acyclic (and symmetric).

2. Preliminaries

Definition 1. [3] An *abstract dialectical framework* (ADF) is a tuple $D = (S, L, C)$ where S is a finite set of nodes (arguments, statements, positions), $L \subseteq S \times S$ is a set of links, and $C = \{\varphi_s\}_{s \in S}$ is a set of propositional formulas (acceptance conditions). Each $\varphi_s \in C$ is constructed out of the *parents* $par(s)$ of $s \in S$, where $par(s) = \{a \in S \mid (a, s) \in L\}$.

We depict ADFs as annotated directed graphs where nodes represent arguments, directed edges represent links, and acceptance conditions are given by the annotations next to arguments. An argument $s \in S$ is called *initial* if $par(s) = \emptyset$. Note that the arguments act as propositional atoms in the acceptance conditions.

Let $D = (S, L, C)$ be an ADF. An *interpretation* v (for D) is a function mapping arguments to one of the truth values true ($\mathbf{t}$), false ($\mathbf{f}$), or undecided ($\mathbf{u}$), i.e. $v : S \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. An interpretation v is *two-valued* if it maps each argument to either $\mathbf{t}$ or $\mathbf{f}$. The sets $v^{\mathbf{t}}$, $v^{\mathbf{f}}$, and $v^{\mathbf{u}}$ contain those arguments that v maps to true, false and undecided, respectively. Further, v is called *trivial*, denoted $v_{\mathbf{u}}$, if $v(s) = \mathbf{u}$ for each $s \in S$. Finally, we denote the update of an interpretation v with a truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ for an argument b by $v|^b_{\mathbf{x}}$, i.e., $v|^b_{\mathbf{x}}(b) = \mathbf{x}$ and $v|^b_{\mathbf{x}}(a) = v(a)$ for $a \neq b$.

Interpretations are ordered w.r.t. their information content. This is based on the partial order of truth values, for which $\mathbf{u} \leq_i \mathbf{t}$ and $\mathbf{u} \leq_i \mathbf{f}$. An interpretation v is at least as informative as another interpretation w, denoted by $w \leq_i v$, if $w(s) \leq_i v(s)$ for each $s \in S$. As usual $w <_i v$ whenever $w \leq_i v$ and not $v \leq_i w$. The *meet* operator $\sqcap_i$ on truth values is defined as $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$ and returns $\mathbf{u}$ otherwise. The meet $\sqcap_i$ of two interpretations v and w is then defined as $(v \sqcap_i w)(s) = v(s) \sqcap_i w(s)$ for each $s \in S$.

Semantics for ADFs are defined based on the *characteristic operator* Γ_D which maps interpretations to interpretations. Given an interpretation v (for D) it is defined as

$$\Gamma_D(v) = v' \text{ such that } v'(s) = \begin{cases} \mathbf{t} & \text{if } \varphi_s^v \text{ is irrefutable (i.e., a tautology) ,} \\ \mathbf{f} & \text{if } \varphi_s^v \text{ is unsatisfiable,} \\ \mathbf{u} & \text{otherwise.} \end{cases}$$

This uses the partial valuation of φ_s by v, which is $\varphi_s^v = \varphi_s[p/\top : v(p) = \mathbf{t}][p/\bot : v(p) = \mathbf{f}]$ where p is an argument occurring in φ_s. The semantics of ADFs are defined via the characteristic operator as in Definition 2.

Definition 2. Given an ADF D, an interpretation v is

- *admissible* in D iff $v \leq_i \Gamma_D(v)$;
- *complete* in D iff $v = \Gamma_D(v)$;
- *grounded* in D iff v is the least fixed-point of Γ_D;
- *preferred* in D iff v is $\leq_i$-maximal admissible (resp. complete);
- a *(two-valued) model* of D iff v is two-valued and $\forall s \in S : v(s) = v(\varphi_s)$;
- a *stable model* of D if v is a model of D and $v^{\mathbf{t}} = w^{\mathbf{t}}$, where w is the grounded interpretation of the *stb*-reduct $D^v = (S^v, L^v, C^v)$, where $S^v = v^{\mathbf{t}}$, $L^v = L \cap (S^v \times S^v)$, and $\varphi_s[p/\bot : v(p) = \mathbf{f}]$ for each $s \in S$.

The set of all σ interpretations for D is denoted by $\sigma(D)$, for $\sigma \in \{adm, com, grd, prf, mod, stb\}$.

In an ADF $D = (S, L, C)$, a link $(b,a) \in L$ is called *supporting* (in D) if for every two-valued interpretation v, $v(\varphi_a) = \mathbf{t}$ implies $v|_{\mathbf{t}}^{b}(\varphi_a) = \mathbf{t}$; a link $(b,a) \in L$ is called *attacking* (in D) if for every two-valued interpretation v, $v(\varphi_a) = \mathbf{f}$ implies $v|_{\mathbf{t}}^{b}(\varphi_a) = \mathbf{f}$. $(b,a) \in L$ is named *redundant* (in D) if it is both attacking and supporting.

An ADF D is called *bipolar* (or BADF for short) if each link in L is attacking in D or supporting in D (thus, the links can also be redundant). An ADF $D = (S, L, C)$ is an AF if the acceptance condition of every argument $s \in S$ is of the form $\varphi_s = \bigwedge_{a \in par(s)} \neg a$. It can then be represented in the more usual form, as a tuple $F = (A, R)$ with arguments $A = S$ and attacks $R = L$.

3. Properties of ADF Subclasses

We begin this section with a novel reformulation of Dung's Fundamental Lemma [2, Lemma 10] in the realm of ADFs. Besides the case of an argument being acceptable (also called defended in AFs) there is the symmetric case where an argument is deniable.

Definition 3. Let $D = (S, L, C)$ be an ADF and v be an interpretation on S. An argument $s \in S$ is called *acceptable* w.r.t. v (in D) if φ_s^v is irrefutable. An argument $s \in S$ is called *deniable* w.r.t. v (in D) if φ_s^v is unsatisfiable.

Lemma 1. *Let v be an admissible interpretation of ADF $D = (S, L, C)$, and a and a' be arguments which are acceptable (resp. deniable) with respect to v in D. Then,*

1. *$v' = v|_{\mathbf{t}}^{a}$ (resp. $v' = v|_{\mathbf{f}}^{a}$) is admissible in D, and*
2. *a' is acceptable (resp. deniable) with respect to v' in D.*

Proof. We show the result for a and a' being acceptable. The deniable case is symmetric.

First note due to φ_a^v being irrefutable and v being admissible, $v(a) \neq \mathbf{f}$. Also, $v'(a) = \mathbf{t}$ by the definition of v'. Hence, $v \leq_i v'$.

(1) We need to show that $v' \leq_i \Gamma_D(v')$, that is, for all $s \in S$, $v'(s) \leq_i \Gamma_D(v')(s)$. Consider an arbitrary $s \in S$. If $v'(s) = \mathbf{u}$ we are done. Assume $v'(s) = \mathbf{t}$. There are two cases

either $s = a$ or $s \neq a$. If $s \neq a$, also $v(s) = \mathbf{t}$. Since v is admissible, φ_s^v is irrefutable. If $s = a$, then s is acceptable w.r.t. v by assumption. Hence, again, φ_s^v is irrefutable. Since we know that $v \leq_i v'$ it follows that also $\varphi_s^{v'}$ is irrefutable. Similarly, we get that $\varphi_s^{v'}$ is unsatisfiable if $v'(s) = \mathbf{f}$. Hence v' is admissible.

(2) By assumption a' is acceptable w.r.t. v, hence $\varphi_{a'}^v$ is irrefutable. Since $v \leq_i v'$, also $\varphi_{a'}^{v'}$ is irrefutable, i.e. a' is acceptable w.r.t. v'. □

3.1. Acyclic ADFs

Definition 4. An ADF $D = (S, L, C)$ is *acyclic* if its corresponding directed graph (S, L) does not contain any directed cycle.

We will need the concept of maximum level of ADFs defined as follows.

Definition 5. The *level* of an argument s of an ADF D is the number of links on the longest path from an initial argument to s plus 1. The *maximum level* of an (acyclic) ADF D is the level of an argument of D that is at least as high as the level of any other argument of D.

It is clear that every acyclic ADF has a maximum level. In the remainder of this section we show that all main semantics coincide for acyclic ADFs.

Proposition 2. *In every acyclic ADF D the $\leq_i$-least fixed point of Γ_D is a model of D.*

Proof. Let $D = (S, L, C)$ be an acyclic ADF and let m be its maximal level. Moreover, let $v_0 := v_{\mathbf{u}}$ and $v_i := \Gamma_D(v_{i-1})$ for $1 \leq i \leq m$. We claim that for all i with $1 \leq i \leq m$, and every argument s_j with level $j \leq i$ it holds that either $v_i(s_j) = \mathbf{t}$ or $v_i(s_j) = \mathbf{f}$. We show this claim by induction on i:

- Base case: Suppose s_1 is an arbitrary argument of level one (an acyclic ADF always includes an initial argument). Since s_1 is an initial argument, either $\varphi_{s_1} = \top$ or $\varphi_{s_1} = \bot$. Hence $v_1(s_1) = \Gamma_D(v_0)(s_1)$ is either true or false.
- Inductive step: Assuming this property holds for all k with $1 \leq k \leq i < m$, we show it holds for $i+1$. We know that $\varphi_{s_j}^{v_i} = \varphi_{s_j}[s_k/\top : v_i(s_k) = \mathbf{t}][s_k/\bot : v_i(s_k) = \mathbf{f}]$. For all s_k that occur in φ_{s_j} it holds that $k < j \leq i+1$, with k being the level of s_k. Therefore, by the inductive hypothesis, for each s_k, either $v_i(s_k) = \mathbf{t}$ or $v_i(s_k) = \mathbf{f}$. Hence either $\varphi_{s_j}^{v_i} \equiv \top$ or $\varphi_{s_j}^{v_i} \equiv \bot$ and, consequently, either $v_{i+1}(s_j) = \mathbf{t}$ or $v_{i+1}(s_j) = \mathbf{f}$.

Since m is the maximum level of any argument in D, we now get that $v_m(s)$ is either true or false for all $s \in S$, i.e. it is a two-valued interpretation. Moreover, it holds that $v_m = \Gamma_D(v_m)$, i.e. v_m is a fixed point.

To show that v_m is the least fixed point of Γ_D, assume, towards a contradiction, that there exists an interpretation $v <_i v_m$ such that $v = \Gamma_D(v)$. Then there exists an argument s such that either $v_m(s) = \mathbf{t}$ or $v_m(s) = \mathbf{f}$, but $v(s) = \mathbf{u}$. Assume s has level i. Since D is an acyclic ADF all arguments s_k that occur in φ_s have a level less than i. Therefore, there exists at least an argument s_j of level $j < i$ in φ_s such that $v(s_j) = \mathbf{u}$. By iterating this method after at most $i-1$ times we reach an argument s_1 of level 1 for which $v(s_1) = \mathbf{u}$. This is a contradiction, since at level 1 it must be the case that either $\varphi_{s_1} = \top$ or $\varphi_{s_1} = \bot$ and therefore $\Gamma_D(v)(s_1) \neq \mathbf{u}$. □

Theorem 3. *In every acyclic ADF D the sets of grounded interpretations, complete interpretations, preferred interpretations, two-valued models, and stable models coincide.*

Proof. First, the grounded interpretation v of D is also complete in D. Moreover, Proposition 2 implies that v is a two-valued model of D. Since $w = w^{\mathbf{t}} = v^{\mathbf{t}}$ in which $w = grd(D^v)$, v is a stable model. It remains to show that there is no further complete interpretation v' of D. Since v is two-valued it must hold that $v \not<_i v'$. However, since v is grounded and therefore the least complete interpretation, such a v' cannot exist. Therefore, v is a unique complete interpretation of D which is grounded, stable, two-valued, and preferred. $\square$

3.2. Symmetric ADFs

The study of symmetric ADFs is motivated by the fact that, as shown in [10], stable and preferred semantics coincide for symmetric AFs, a property called coherence. Due to the distinction between two-valued and stable models in ADFs, we actually consider different levels of coherence.

Definition 6. An ADF D is called

- *coherent* if each preferred interpretation of D is a stable model of D;
- *weakly coherent* if each two-valued model of D is a stable model of D;
- *semi-coherent* if each preferred interpretation of D is a two-valued model of D.

Another concept studied in [10] is relatively groundedness, which is shown to hold for every symmetric AF there. That is, the grounded extension coincides with the intersection of all preferred extensions. We generalize this definition to ADFs.

Definition 7. An ADF D is called *relatively grounded* if $grd(D) = \bigsqcap_i prf(D)$.

Symmetric ADFs are now defined as follows.

Definition 8. An ADF $D = (S, L, C)$ is *symmetric* if L is irreflexive and symmetric and L does not contain any redundant links.

It turns out that neither of the properties analogous to those holding for symmetric *AFs* hold for symmetric *ADFs*.

Theorem 4. *The class of symmetric ADFs is neither semi-coherent, nor weakly coherent, nor relatively grounded.*

Proof. Let D be the symmetric ADF depicted in Figure 1. It holds that $prf(D) = \{v_1, v_2\}$ with $v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{t}, e \mapsto \mathbf{f}\}$ and $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{u}, e \mapsto \mathbf{u}\}$. Since v_1 is a two-valued model of D which is not stable (since $D^v = D$ and $grd(D) = \{v_{\mathbf{u}}\}$), D is not weakly coherent. Also, D is not semi-coherent since v_2 is not two-valued. In addition, $\bigsqcap_i prf(D) = v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}, d \mapsto \mathbf{u}, e \mapsto \mathbf{u}\}$, but $grd(D) = \{v_{\mathbf{u}}\}$. Therefore, D is not relatively grounded. $\square$

This raises the question whether there is a particular subclass of symmetric ADFs which fulfills the properties considered in Theorem 4. We investigate if this is the case for acyclic support symmetric ADFs.

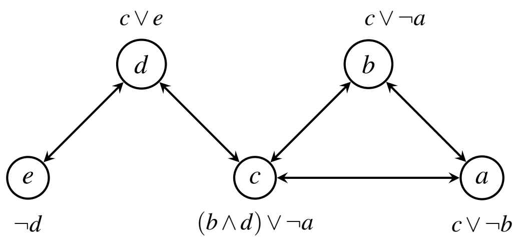


Figure 1. Symmetric ADF which is neither semi-coherent, weakly coherent nor relatively grounded.

Definition 9. Given an ADF $D = (S,L,C)$, let T be the set of all supporting links. The corresponding graph (S,T) is named the *support-reduct* of D. D contains a *support cycle* whenever the support-reduct of D contains a directed cycle.

Definition 10. An ADF D is called *acyclic support symmetric ADF* (ASSADF for short) if it is symmetric, bipolar and does not contain any support cycle.

We begin by showing that ASSADFs are weakly coherent, using the following technical lemma.

Lemma 5. *Let D be an ADF, v be a two-valued model of D, and $s \in S$ be an argument s.t. all parents of s are attackers and s does not occur in φ_s. If φ_s^v is irrefutable then $\varphi_s[s_i/\bot : v(s_i) = \mathbf{f}]$ is irrefutable.*

Theorem 6. *Every acyclic support symmetric ADF is weakly coherent.*

Proof. Let $D = (S,L,C)$ be an acyclic support symmetric ADF. We have to show that each two-valued model of D is also a stable model of D. Let v be a two-valued model of D, $D^v = (S^v, L^v, C^v)$ be the *stb*-reduct of D, w be the unique grounded interpretation of D^v, and φ'_s the acceptance condition of s in D^v, i.e. $\varphi'_s = \varphi_s[s_i/\bot : v(s_i) = \mathbf{f}]$. We show that $v^{\mathbf{t}} = w^{\mathbf{t}}$. Suppose to the contrary that there exists an argument s, s.t. $v(s) = \mathbf{t}$ and $w(s) \neq \mathbf{t}$. That is, $\varphi_s'^v \not\equiv \top$. This means, by Lemma 5 that $\varphi_s'^v$ contains an argument s_1 supporting s. For s_1 to appear in $\varphi_s'^v$ it must be that $v(s_1) = \mathbf{t}$. Since supports are acyclic in ASSADFs, by the same reason $\varphi'_{s_1} = \varphi_{s_1}[s_i/\bot : v(s_i) = \mathbf{f}]$ contains an argument s_2 which is different from s and s_1 and which supports s_1. Thus there exists an infinite sequence of $s_1, s_2, \ldots$ s.t. s_{i+1} supports s_i. This is a contradiction to D being an ASSADF. $\square$

On the other hand, ASSADFs are neither semi-coherent nor relatively grounded.

Theorem 7. *The class of ASSADFs is neither semi-coherent nor relatively grounded. Even ASSADFs without supporting links are not semi-coherent.*

Proof. Consider the ASSADF D depicted in Figure 2. D has 4 preferred interpretations, namely $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f},\ c \mapsto \mathbf{t},\ d \mapsto \mathbf{t},\ e \mapsto \mathbf{f}\}$, $v_2 = \{a \mapsto \mathbf{f},\ b \mapsto \mathbf{t},\ c \mapsto \mathbf{f},\ d \mapsto \mathbf{t},\ e \mapsto \mathbf{f}\}$, $v_3 = \{a \mapsto \mathbf{t},\ b \mapsto \mathbf{f},\ c \mapsto \mathbf{f},\ d \mapsto \mathbf{t},\ e \mapsto \mathbf{f}\}$, and $v_4 = \{a \mapsto \mathbf{u},\ b \mapsto \mathbf{u},\ c \mapsto \mathbf{u},\ d \mapsto \mathbf{f},\ e \mapsto \mathbf{t}\}$. As every two-valued interpretation of D (that is v_1, v_2 and v_3) is also a stable model, D is weakly coherent, confirming Theorem 6. However, v_4 is a preferred interpretation which is not a two-valued model. Hence, D is not semi-coherent.

We show that ASSADFs are not relatively grounded. Consider the ASSADF $D = (S,L,C)$ with $S = \{a,b,c\}$, $\varphi_a : \neg b \wedge \neg c$, $\varphi_b : \neg a \wedge \neg c$, and $\varphi_c : a \vee \neg b$. D has preferred

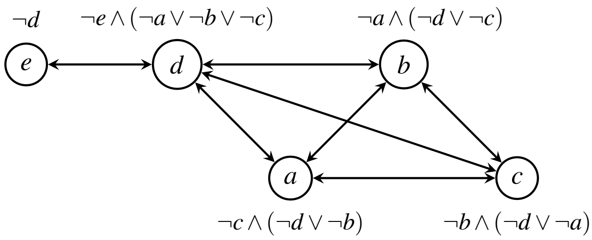


Figure 2. ASSADF without supporting links being not semi-coherent.

interpretations $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}\}$ and $v_2 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}$. We obtain $v_1 \sqcap_i v_2 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}$. However, the grounded interpretation of D is the trivial interpretation v_u. That is, D is not relatively grounded. $\square$

4. Expressiveness of ADF Subclasses

In this section we deal with the expressiveness of formalisms from the perspective of realizability [13]. In particular, we are interested in how the novel class of ASSADFs behaves in these terms compared to AFs and (B)ADFs. The relationship between AFs, BADFs, and ADFs under the main semantics has been studied in previous work [15,16].

A set of interpretations V is said to be σ-realizable in a formalism $\mathscr{F}$, which is the set of structures available in $\mathscr{F}$, if there exists an element kb ("knowledge base") of $\mathscr{F}$ s.t. $\sigma(\mathrm{kb}) = V$. The *signature* $\Sigma^{\sigma}_{\mathscr{F}}$ of a formalism $\mathscr{F}$ is then defined as follows.

Definition 11. The *signature* $\Sigma^{\sigma}_{\mathscr{F}}$ of a formalism $\mathscr{F}$ w.r.t. a semantics σ is defined as:

$$\Sigma^{\sigma}_{\mathscr{F}} = \{\sigma(\mathrm{kb}) \mid \mathrm{kb} \in \mathscr{F}\}.$$

Given two formalisms $\mathscr{F}_1$ and $\mathscr{F}_2$, we say that $\mathscr{F}_1$ is *strictly more expressive* than $\mathscr{F}_2$ for σ, whenever $\Sigma^{\sigma}_{\mathscr{F}_2} \subsetneq \Sigma^{\sigma}_{\mathscr{F}_1}$. $\mathscr{F}_1$ and $\mathscr{F}_2$ are incomparable under semantics σ if neither $\Sigma^{\sigma}_{\mathscr{F}_1} \subseteq \Sigma^{\sigma}_{\mathscr{F}_2}$ nor $\Sigma^{\sigma}_{\mathscr{F}_2} \subseteq \Sigma^{\sigma}_{\mathscr{F}_1}$. This is denoted as $\Sigma^{\sigma}_{\mathscr{F}_1} \not\sim \Sigma^{\sigma}_{\mathscr{F}_2}$.

In the following, we will study the signature of ASSADFs w.r.t. the semantics under consideration, compared to AFs and BADFs. We begin by showing that BADFs are strictly more expressive than ASSADFs for $\sigma \in \{adm, prf, com, mod\}$.

Theorem 8. *For $\sigma \in \{adm, prf, com, mod\}$ it holds that $\Sigma^{\sigma}_{ASSADF} \subsetneq \Sigma^{\sigma}_{BADF}$.*

Proof. Since every ASSADF is, by definition, a BADF, $\Sigma^{\sigma}_{\mathrm{ASSADF}} \subseteq \Sigma^{\sigma}_{\mathrm{BADF}}$ is clear. To show that $\Sigma^{\sigma}_{\mathrm{BADF}}$ is a strict superset of $\Sigma^{\sigma}_{\mathrm{ASSADF}}$ it is enough to find a set of interpretations V which is σ-realizable in BADFs, but not σ-realizable in ASSADFs.

For $\sigma \in \{prf, mod\}$, let $V = \{\{a \mapsto \mathbf{t}\}, \{a \mapsto \mathbf{f}\}\}$, and for $\sigma' \in \{com, adm\}$, let $V' = \{\{a \mapsto \mathbf{u}\}, \{a \mapsto \mathbf{t}\}, \{a \mapsto \mathbf{f}\}\}$. The BADF $D = (S, L, C)$ with $S = \{a\}$ and $\varphi_a = a$ realizes V under σ and V' under σ'. On the other hand, it is easy to check that there is no ASSADF with one argument which realizes V under σ, and respectively, V' under σ'. (If there is an ASSADF with one argument a then either $\varphi_a = \top$ or $\varphi_a = \bot$. Thus, it can realize neither V under σ nor V' under σ'.) $\square$

In the proof of Theorem 9 we show why, on the other hand, AFs and ASSADFs are incomparable for $\sigma \in \{adm, prf, com\}$.

Theorem 9. $\Sigma^{\sigma}_{AF} \not\sim \Sigma^{\sigma}_{ASSADF}$*, for* $\sigma \in \{adm, prf, com\}$.

Proof. To demonstrate that AFs and ASSADFs are incomparable under $\sigma \in \{adm, prf, com\}$ we show that $\Sigma^{\sigma}_{\mathrm{AF}} \not\subseteq \Sigma^{\sigma}_{\mathrm{ASSADF}}$ and $\Sigma^{\sigma}_{\mathrm{ASSADF}} \not\subseteq \Sigma^{\sigma}_{\mathrm{AF}}$.

• To show $\Sigma^{\sigma}_{\mathrm{AF}} \not\subseteq \Sigma^{\sigma}_{\mathrm{ASSADF}}$ consider $V = \{\{a \mapsto \mathbf{u}\}\}$. A witness of σ-realizability in AFs is $F = (\{a\}, \{(a,a)\})$. However, there is no ASSADF to realize V under σ.

• To verify that $\Sigma^{\sigma}_{\mathrm{ASSADF}} \not\subseteq \Sigma^{\sigma}_{\mathrm{AF}}$ for $\sigma \in \{adm, prf, com\}$ we first show that $\Sigma^{prf}_{\mathrm{ASSADF}} \not\subseteq \Sigma^{prf}_{\mathrm{AF}}$. Let $V = \{v_1, v_2, v_3\}$ with $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, e \mapsto \mathbf{t}\}$, $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{t}, e \mapsto \mathbf{f}\}$, and $v_3 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}, e \mapsto \mathbf{f}\}$. A witness of *prf*-realizability of V in ASSADFs is $D = (S, L, C)$ with $S = \{a, b, c, e\}$, $\varphi_a = \neg e \wedge (\neg b \vee \neg c)$, $\varphi_b = \neg a \vee \neg c$, $\varphi_c = \neg a \vee \neg b$, and $\varphi_e = \neg a$. However, there is no AF with V as its preferred interpretations. (If there is an AF F' s.t. $\sigma(F') = V$ then the structure of v_1, v_2 and v_3 implies that there is no attack between a, b and c in F'. Thus, if there is an attack from any of a, b and c to e then $\{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, e \mapsto \mathbf{f}\}$ is a preferred interpretation of F'. If there is no attack from any of a, b and c to e then $\{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, e \mapsto \mathbf{t}\}$ is a preferred interpretation of F'. In both cases $\sigma(F') \neq V$.) For $\sigma = com$ let $V' = V \cup \{\{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}, e \mapsto \mathbf{u}\}\}$. It is easy to check that V' is *com*-realizable by the ASSADF D defined above. If there is an AF F' that realizes V' under *com* then each of the elements of V would be a preferred interpretation of F'. Thus, $prf(F') = V$ would be the case, which it is easy to see is actually false. Finally, we get $\Sigma^{adm}_{\mathrm{ASSADF}} \not\subseteq \Sigma^{adm}_{\mathrm{AF}}$ by observing that from $adm(D)$ being realizable under *adm* in AFs it would follow that $prf(D)$ is realizable under *prf* in AFs. But we already showed that the latter is not the case. □

Theorems 8–9 together with results from [16] are summarized in Figure 3. The situation is different for the stable semantics. We recall that stable models v, w of an ADF are always incomparable, i.e. $w^{\mathbf{t}} \subseteq v^{\mathbf{t}}$ implies $w^{\mathbf{t}} = v^{\mathbf{t}}$, see [17].

Theorem 10. $\Sigma^{stb}_{AF} \subsetneq \Sigma^{stb}_{ASSADF} \sim \Sigma^{stb}_{BADF}$.

Proof. (Sketch) $\Sigma^{stb}_{\mathrm{AF}} \subsetneq \Sigma^{stb}_{\mathrm{BADF}}$ is shown in [15]; if we can show $\Sigma^{stb}_{\mathrm{ASSADF}} \sim \Sigma^{stb}_{\mathrm{BADF}}$ we are thus done. We do so by showing that any incomparable set of two-valued interpretations is *stb*-realizable by some ASSADF. To this end, let V be an incomparable set over arguments S (i.e., each $v \in V$ assigns $\mathbf{t}$ or $\mathbf{f}$ to all $s \in S$) and consider an ASSADF $D = (S, L, C)$ with the following acceptance conditions for $s \in S$[2]:

- If $v(s) = \mathbf{t}$ for every $v \in V$ then $\varphi_s = \top$.
- If $v(s) = \mathbf{f}$ for every $v \in V$ then $\varphi_s = \bot$.
- Otherwise, $\varphi_s = \bigvee_{v \in V, v(s) = \mathbf{t}} \quad \bigwedge_{v(t) = \mathbf{f} \wedge \exists w \in V : (w(s) = \mathbf{f} \wedge w(t) = \mathbf{t})} \neg t$.

By definition, all links in D are attacking and symmetry follows by construction and the fact that V is incomparable. Moreover, it can be shown that the stable models of D are exactly given by V. □

[2] In fact, this construction is a slight adaption of a result in [15].

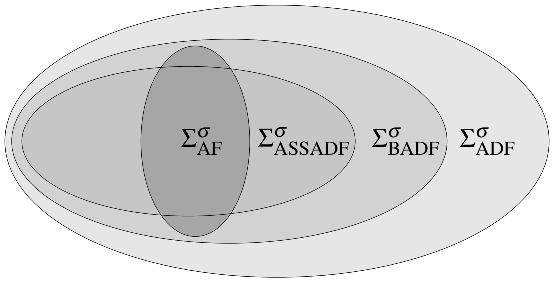


Figure 3. Expressiveness of subclasses of ADFs for $\sigma \in \{adm, prf, com\}$

The following is a consequence of Theorem 10, the fact that *stb* and *mod* are equivalent for AFs, and that ASSADFs are weakly coherent (cf. Theorem 6), and $\Sigma^{mod}_{\mathrm{ASSADF}} \subsetneq \Sigma^{mod}_{\mathrm{BADF}}$ from Theorem 8.

Corollary 11. $\Sigma^{mod}_{AF} \subsetneq \Sigma^{mod}_{ASSADF} \subsetneq \Sigma^{mod}_{BADF}$.

The picture changes when we restrict the cardinality of interpretation sets. As it turns out, any set of interpretations of size 2 obtained from an ADF under the stable semantics is also realizable in AFs.

Proposition 12. *Suppose that $|V| = 2$ and V is stb-realizable in ADFs. Then V is stb-realizable in AFs.*

Proof. Let $V = \{v_1, v_2\}$ be a set of interpretation that is *stb*-realizable in ADFs, i.e. there there exists an ADF $D = (S, L, C)$ s.t. $stb(D) = V$. Construct an AF $F = (A, R)$ by setting $A = S$ and $R = \{(a, b) \mid v_i(a) = \mathbf{t}, v_i(b) = \mathbf{f}, v_j(a) = \mathbf{f}, 1 \leq i \neq j \leq 2\}$. To prove that $stb(F) = V$, take $v_i \in V$. First, there is no attack between arguments with $v_i(a) = \mathbf{t}$. Moreover, if $v_i(b) = \mathbf{f}$ then, since neither $v_1 \leq_i v_2$ nor $v_2 \leq_i v_1$, there must be some $a \in A$ with $v_i(a) = \mathbf{t}$ and $v_j(a) = \mathbf{f}$, hence this $(a, b) \in R$. Hence v_i is a stable interpretation of F. That is, V is *stb*-realizable in AFs. $\square$

5. Experiments

We turn now to the experimental section of our work, where we report on results on an initial investigation of the performance of solvers for ADFs on subclasses of ADFs. We focused on the question to what extent current systems are tuned to the acyclic vs. non-acyclic nature of ADFs. We carried out experiments for credulous acceptance w.r.t. the admissible semantics and skeptical acceptance w.r.t. the preferred semantics. From Theorem 3 and complexity results for ADFs [9] it follows that these reasoning tasks can be decided in polynomial time for acyclic ADFs, while they are located on the second and third level of the polynomial hierarchy respectively in general.

All systems for ADFs we are aware of rely on encodings to some other formalism; namely, to quantified Boolean formulas (QBFs) and answer set programming (ASP). None of these systems are tailored specifically to acyclic ADFs. Therefore, the question becomes to what extent the solvers used at the backend of current ADF systems are able to use the acyclic nature of ADFs in their favour.

To answer the question at hand we have modified an existing generator for ADFs (used in [18]) to take an undirected graph as input[3] and be able to generate an acyclic as well as a non-acyclic ADF as output. The generated ADF inherits the structure of the graph; vertices in the graph become arguments in the ADF. As to the links of the ADF, in the case of an acyclic ADF a directed graph is first generated from the undirected graph by choosing a total order on the vertices at random. This total order is used to determine the direction of the links. For the non-acyclic ADF, a probability controls whether an edge in the input graph will result in a symmetric link in the ADF (we used a probability of 0.5); in case of non-symmetric links the direction of the link is chosen at random.

Having the links of the ADF, the procedure of generating the acceptance conditions is the same for acyclic as well as non-acyclic ADFs. Each parent of an argument is assigned to one of 5 different groups (with equal probability), determining whether the parent participates in a subformula of the argument's acceptance condition representing the notions of attack, group-attack, support, or group-support familiar from argumentation. Also, the parents can appear as literals connected by XOR (to capture the full complexity of ADFs)[4]. The subformulas are connected via $\wedge$ or $\vee$ with equal probability.

In our experiments the input graphs stem from data-sets used at the second international competition on computational models of argumentation (ICCMA)[5], namely from encoding assumption-based argumentation problems into AFs ("ABA"), encoding planning problems as AFs ("Planning"), and a data-set of AFs generated from traffic networks ("Traffic"). More specifically, based on preliminary experiments, we selected 100 AFs at random from a subset of AFs having up to 150 arguments in the very dense AFs in the "ABA" data-set, and 100 AFs at random from a subset of AFs having up to 300 arguments in each of the "Planning" and "Traffic" benchmarks. From the resulting 300 AFs interpreted as undirected graphs, we generated 300 acyclic and 300 non-acyclic ADFs.

We evaluated the performance of the QBF-based system `QADF` [19] as well as the ASP-based systems `goDIAMOND` [20] and `YADF` [18]. We studied the running times for credulous reasoning for the admissible and skeptical reasoning for the preferred semantics. We used a 48 GB Debian (8.5) machine with 8 Intel Xeon processors (2.33 GHz). For `QADF` we used version 0.3.2 together with the preprocessor `bloqqer` 035 and the QSAT-solver `DepQBF` 4.0. `YADF` is version 0.1.0 with the rule decomposition tool `lpopt` and the ASP-solver `clingo` 4.4.0. We gave the systems 10 minutes computation time.

Table 1 and Figure 4 summarise the results of our experiments. For the admissible semantics, the interesting result is the significant improvement of `QADF` on the acyclic ADFs. For `goDIAMOND` there seems also to be a slight improvement on the "Planning" and "Traffic" benchmarks (although two time-outs[6] on the acyclic instances of "Planning"). For `YADF` there is some improvement on the acyclic "ABA" benchmarks.

In the case of the preferred semantics, `QADF` timed-out on most acyclic as well as non-acyclic instances (in particular, all of the "Planning" instances). `YADF` also had quite a few time-outs, especially on the "ABA" and "Planning" benchmarks on which other-

[3] Note that the version reported on in [18] takes a directed graph as input.

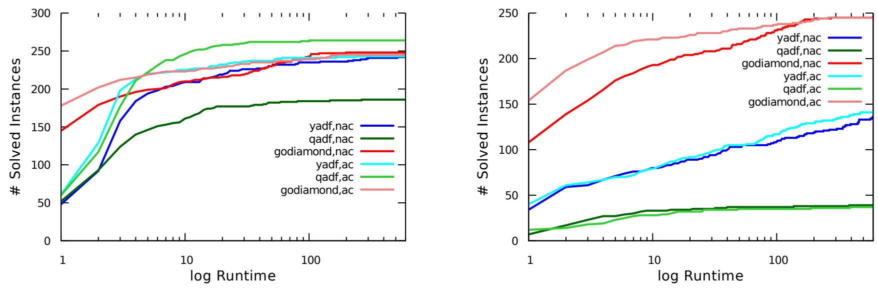
[4] In order not to bias the results of the experiments artificially in favour of solvers that are able to process acceptance conditions with XOR without first having to simplify them, the parents appear in groups of up to 5 connected by XOR. These subformulas are, in turn, connected with $\wedge$ or $\vee$ with equal probability.

[5] See `http://argumentationcompetition.org/2017/index.html`.

[6] In the case of `goDIAMOND` there were actually almost no time-outs, yet memory-errors. For simplicity, we do not distinguish between time-outs and memory-errors here.

		Credulous-admissible				Skeptical-preferred			
		Acyclic		Non-acyclic		Acyclic		Non-acyclic	
		Time-outs	Mean	Time-outs	Mean	Time-outs	Mean	Time-outs	Mean
ABA	goDIAMOND	52	21.26	52	22.64	52	27.58	52	27.67
	QADF	23	2.96	30	8.15	83	14.79	81	32.73
	YADF	54	7.38	56	31.39	54	16.77	57	39.46
Planning	goDIAMOND	2	4.27	0	7.678	2	8.66	0	17.03
	QADF	11	6.07	59	14.63	100	NA	100	NA
	YADF	2	8.13	0	13.20	70	124.22	71	202.93
Traffic	goDIAMOND	1	0.74	1	4.92	1	1.19	3	11.12
	QADF	2	3.07	25	2.15	80	27.79	80	4.7
	YADF	1	1.65	2	5.68	35	42.18	36	30.57

Table 1. Time-outs and mean running times (in seconds) for all solvers. Mean running times are computed disregarding time-outs. Number of time-outs are always out of 100 ADF instances.



(a) Credulous-admissible

(b) Skeptical-preferred

Figure 4. Plot of running time (in seconds) vs. number of instances solved (out of all benchmarks) for all ADF solvers. “ac” stands for “acyclic”, “nac” for “non-acyclic”.

wise the system fared better on the acyclic instances. `goDIAMOND` performed much better than both other systems (still timing-out on around half of the acyclic as well as non-acyclic “ABA” instances), and there is again some improvement in the performance on the acyclic vs. non-acyclic instances of the “Planning” and “Traffic” benchmarks.

6. Discussion

In this work, we introduced several subclasses of ADFs and investigated their properties. We showed that acyclic ADFs behave similar to acyclic AFs in terms of different semantics. We discussed that for symmetric ADFs the picture is different and related the expressibility of a particular class of symmetric ADFS (ASSADFs) to the expressibility of AFs and BADFs. We also provided experiments to analyse the performance of solvers when applied to these subclasses. Our results show that some systems are more efficient on acyclic instances (compared to cyclic ones of similar size), while others are not, which indiciates possible optimization potential for the latter (the complexity of acyclic ADFs has been recently adressed in [21]). As future work, we also consider to extend our studies to other ADF semantics [22,23].

Acknowledgements. This research has been supported by FWF projects I2854, P30168, and W1255. The second researcher is currently embedded in the Center of Data Science & System Complexity (DSSC) Doctoral Programme, at the University of Groningen.

References

[1] Atefeh Keshavarzi Zafarghandi. Investigating subclasses of abstract dialectical frameworks. Master's thesis, TU Wien, 2017.

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

[3] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proc. KR*, pages 102–111, 2010.

[4] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran. Abstract Dialectical Frameworks. An Overview. *IfCoLog Journal of Logics and their Applications*, 4(8):2263–2318, 2017.

[5] Elena Cabrio and Serena Villata. Abstract dialectical frameworks for text exploration. In *Proc. ICAART*, pages 85–95, 2016.

[6] Jörg Pührer. ArgueApply: A mobile app for argumentation. In *Proc. LPNMR*, pages 250–262. Springer, 2017.

[7] Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. A methodology for designing systems to reason with legal cases using abstract dialectical frameworks. *Artif. Intell. Law*, 24(1):1–49, 2016.

[8] Daniel Neugebauer. Generating defeasible knowledge bases from real-world argumentations using D-BAS. In *Proc. AI^3@AI*IA 2017*, volume 2012, pages 105–110. CEUR Workshop Proceedings, 2017.

[9] Hannes Strass and Johannes P. Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artif. Intell.*, 226:34–74, 2015.

[10] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *Proc. ECSQARU*, pages 317–328. Springer, 2005.

[11] Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.

[12] Wolfgang Dvořák, Matti Järvisalo, Johannes P. Wallner, and Stefan Woltran. Complexity-sensitive decision procedures for abstract argumentation. *Artif. Intell.*, 206:53–78, 2014.

[13] Paul E. Dunne, Wolfgang Dvorák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228:153–178, 2015.

[14] Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proc. IJCAI*, pages 3171–3177. AAAI Press, 2015.

[15] Hannes Strass. Expressiveness of two-valued semantics for abstract dialectical frameworks. *J. Artif. Intell. Res.*, 54:193–231, 2015.

[16] Thomas Linsbichler, Jörg Pührer, and Hannes Strass. A uniform account of realizability in abstract argumentation. In *Proc. ECAI*, pages 252–260. IOS Press, 2016.

[17] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.*, 205:39–70, 2013.

[18] Gerhard Brewka, Martin Diller, Georg Heissenberger, Thomas Linsbichler, and Stefan Woltran. Solving advanced argumentation problems with answer-set programming. In *Proc. AAAI*, pages 1077–1083. AAAI Press, 2017.

[19] Martin Diller, Johannes Peter Wallner, and Stefan Woltran. Reasoning in abstract dialectical frameworks using quantified boolean formulas. In *Proc. COMMA*, pages 241–252. IOS Press, 2014.

[20] Hannes Strass and Stefan Ellmauthaler. goDIAMOND 0.6.6 – ICCMA 2017 System Description, 2017. Available at `http://argumentationcompetition.org/2017/goDIAMOND.pdf`.

[21] Thomas Linsbichler, Marco Maratea, Andreas Niskanen, Johannes P. Wallner, and Stefan Woltran. Novel algorithms for abstract dialectical frameworks based on complexity analysis of subclasses and SAT solving. In *Proc. IJCAI*, 2018. To appear.

[22] Sarah Alice Gaggl, Sebastian Rudolph, and Hannes Strass. On the computational complexity of naive-based semantics for abstract dialectical frameworks. In *Proc. IJCAI*, pages 2985–2991. IJCAI/AAAI, 2015.

[23] Sylwia Polberg. Understanding the abstract dialectical framework. In *Proc. JELIA*, volume 10021 of *Lecture Notes in Computer Science*, pages 430–446, 2016.