Bipolar Abstract Dialectical Frameworks Are Covered by Kleene's Three-valued Logic

Ringo Baumann[1,2] , **Maximilian Heinrich**[3]
[1]Computer Science Institute, Leipzig University, Germany
[2]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig, Germany
[3]Intelligent Information Systems, Bauhaus-Universität Weimar, Germany
baumann@informatik.uni-leipzig.de, maximilian.heinrich@uni-weimar.de

Abstract

Abstract dialectical frameworks (ADFs) are one of the most powerful generalizations of classical Dung-style argumentation frameworks (AFs). The additional expressive power comes with an increase in computational complexity, namely one level up in the polynomial hierarchy in comparison to their AF counterparts. However, there is one important subclass, so-called bipolar ADFs (BADFs) which are as complex as classical AFs while offering strictly more modeling capacities. This property makes BADFs very attractive from a knowledge representation point of view and is the main reason why this class has received much attention recently. The semantics of ADFs rely on the Γ-operator which takes as an input a three-valued interpretation and returns a new one. However, in order to obtain the output the original definition requires to consider any two-valued completion of a given three-valued interpretation. In this paper we formally prove that in case of BADFs we may bypass the computationally intensive procedure via applying Kleene's three-valued logic $\mathcal{K}_3$. We therefore introduce the so-called bipolar disjunctive normal form which is simply a disjunctive normal form where any used atom possesses either a positive or a negative polarity. We then show that: First, this normal form is expressive enough to represent any BADF and secondly, the computation can be done via Kleene's $\mathcal{K}_3$ instead of dealing with two-valued completions. Inspired by the main correspondence result we present some first experiments showing the computational benefit of using Kleene.

1 Introduction

The field of computational models of argument has become a vibrant research area in Artificial Intelligence (AI) [Bench-Capon and Dunne, 2007; Atkinson *et al.*, 2017; Baroni *et al.*, 2018]. One main reason for the recently increased interest is the insight that broad acceptance for AI technologies can only be attained if recommended decisions are *explainable* to the user. That is, especially in areas with far-reaching effects like criminal justice, finance sector or healthcare, we want trustworthy systems which are able to respond why a certain option was (not) chosen and to react on counterarguments in an interactive way. Computational models of argumentation are engaged with modeling arguments and their relationships, as well as the evaluation of conflicting scenarios. Consequently, they are perfectly suited for such explanation tasks and thus might be used as an additional component for an AI system [Cocarascu and Toni, 2016].

One can distinguish two major lines of research in the field: logic-based and abstract approaches. The former takes the logical structure of arguments into account and defines notions like attack, undercut, defensibility etc. in terms of logical properties of the chosen argument structures (cf. [Besnard and Hunter, 2008] for an excellent overview). At the heart of the second approach are currently Dung's widely used *abstract argumentation frameworks* (AFs) and their associated semantics [Dung, 1995]. Through the years the community realized that the limited expressive capability of AFs, namely the option of single attacks only, reduces their suitability as sound target systems for more complex applications [Atkinson *et al.*, 2017]. Therefore a number of additional functionalities were introduced encompassing preferences, values, collective attacks, attacks on attacks as well as support relations between arguments [Amgoud and Vesic, 2011; Bench-Capon and Atkinson, 2009; Nielsen and Parsons, 2006; Baroni *et al.*, 2009; Cayrol and Lagasquie-Schiex, 2009]. One of the most powerful generalizations of Dung AFs, yet staying on the abstract layer, are *abstract dialectical frameworks* (ADFs) [Brewka and Woltran, 2010; Brewka *et al.*, 2014]. The additional expressive power is achieved by adding acceptance conditions to the arguments which allow for the specification of more complex relationships between arguments, e.g. collective attacks as well as single and collective support.

Semantics for ADFs generalize classical semantics and coincide with them if Dung-style acceptance functions are used. They rely on different (pre-)fixpoints of the so-called Γ-*operator* and in general, their computational complexity is one level up in the polynomial hierarchy compared to their AF counterparts [Straß and Wallner, 2015]. Interestingly, there is the subclass of *bipolar ADFs* (BADFs) which are as complex as AFs while arguably offering more modeling capacities [Brewka *et al.*, 2017b]. This property makes BADFs very attractive from a knowledge representation point of view and is the reason why this class has received much attention recently [Alviano *et al.*, 2016; Baumann and Straß, 2017; Straß, 2018]. In this paper we make further contributions to this line of research, namely:

1. Introducing a new class of propositional formulae, so-called *bipolar disjunctive normal forms* (bipolar DNFs), and showing that bipolar ADFs are representable via these normal forms.
2. We then proceed with the main theorem showing that *Kleene's three-valued logic* $\mathcal{K}_3$ [Kleene, 1952] is suitable for bipolar DNFs allowing to bypass the computationally intensive procedure of the Γ-operator.
3. Extending the main result to *syntactically bipolar formulae* [Straß, 2015] which offer more representational freedom.
4. Implementing a solver and conducting experiments showing the computational benefit of using Kleene's $\mathcal{K}_3$.

2 Background

2.1 Propositional Logic and Kleene's $\mathcal{K}_3$

In the following we recap standard logical concepts. Both logics use the same syntax. Let A be a fixed set of atoms, then the set of formulae $\mathcal{F}$ (over A) is inductively defined as: 1. $A \subseteq \mathcal{F}$ and 2. If $\phi, \psi \in \mathcal{F}$, then $\phi \wedge \psi, \phi \vee \psi, \neg\phi \in \mathcal{F}$. We consider two-valued resp. three-valued interpretations v which assign one of the truth values *true* ($\mathbf{t}$), *false* ($\mathbf{f}$) or *unknown* ($\mathbf{u}$) to each atomic formula. Formally, $v : A \to \{\mathbf{t}, \mathbf{f}\}$ or $v : A \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, respectively. We use $\mathcal{V}_2$ ($\mathcal{V}_2^A$) and $\mathcal{V}_3$ ($\mathcal{V}_3^A$) for the set of all two- resp. three-valued interpretations (over A). Kleene's $\mathcal{K}_3$ generalizes classical logic as it allows to assign the truth value unknown [Kleene, 1952; Fitting, 1991]. However, in the classical corner cases both logics coincide. The associated truth tables are given in Table 1. We use $w(\phi)$ and $v^{\mathcal{K}_3}(\phi)$ to indicate that formula ϕ is evaluated w.r.t. classical logic or Kleene, respectively. Consequently, we implicitly have that $w \in \mathcal{V}_2$ and $v \in \mathcal{V}_3$.

a	b	$a \vee b$	$a \wedge b$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$
$\mathbf{t}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$
$\mathbf{t}$	$\mathbf{u}$	$\mathbf{t}$	$\mathbf{u}$
$\mathbf{f}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{f}$
$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
$\mathbf{f}$	$\mathbf{u}$	$\mathbf{u}$	$\mathbf{f}$
$\mathbf{u}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{u}$
$\mathbf{u}$	$\mathbf{f}$	$\mathbf{u}$	$\mathbf{f}$
$\mathbf{u}$	$\mathbf{u}$	$\mathbf{u}$	$\mathbf{u}$

a	$\neg a$
$\mathbf{t}$	$\mathbf{f}$
$\mathbf{f}$	$\mathbf{t}$
$\mathbf{u}$	$\mathbf{u}$

Table 1: Kleene's three-valued logic $\mathcal{K}_3$

2.2 Abstract Dialectial Frameworks (ADFs)

In a nutshell, ADFs are just directed graphs equipped with further acceptance information. We briefly recall some notation and refer to [Brewka *et al.*, 2017b] for an exhaustive overview. An ADF is a triple $D = (S, L, C)$ where $S \subseteq A$ is a set of statements, $L \subseteq S \times S$ is a set of links, and $C = \{C_s | s \in S\}$ is a set of acceptance functions with $C_s : 2^{par(s)} \to \{\mathbf{t}, \mathbf{f}\}$ where $par(s) = \{s' \mid (s', s) \in L\}$ denotes the set of parent nodes of s. Each C_s precisely specifies when a statement s is getting accepted. It is common to represent an acceptance function C_s as a propositional (acceptance) formula ϕ_s over the vocabulary $par(s)$ (cf. [Straß, 2015] for more details).[1] In this case we write $D = (S, \Phi)$ with $\Phi = \{\phi_s \mid s \in S\}$ and leave the links implicit. It is important to have in mind that the latter representation is not uniquely determined as syntactically different formulae may represent the same acceptance function. Let us proceed with an example.

Example 1. *Suppose that Jack has gotten the offer for a new job (j). He would surely accept this offer if the new position offers a high salary (h). Alternatively Jack would take the job if it provides a meaningful activity for society (m) and he has the opportunity to work remotely. It is known that the job is meaningful and he does not have to be physically at his workplace (w). However, the salary has yet to be discussed. The associated ADF can be given as* $D = (\{j, h, m, w\}, \{\phi_j = h \vee (m \wedge \neg w), \phi_h = h, \phi_m = \top, \phi_w = \bot\})$.

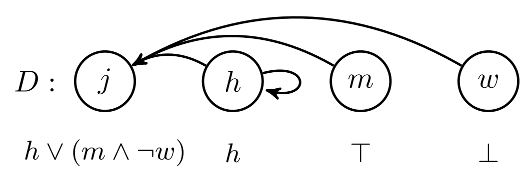

Figure 1: Jack's job offer

2.3 Γ-Operator and Semantics

Argumentation semantics rely on computing the so-called Γ-*operator*. In order to present this operator we have to introduce the so-called *information order* $<_i$ which assigns a greater informational content of the classical truth values, i.e. $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ (cf. [Straß and Wallner, 2015] for more background information). We use $\leq_i$ for the reflexive closure of $<_i$. The resulting meet-operation $\sqcap_i$ on $(\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}, \leq_i)$ is called *consensus* which assigns $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$, and $\mathbf{u}$ otherwise. Finally, we generalize $\leq_i$ to three-valued interpretations in the following point-wise way: $v_1 \leq_i v_2$ iff for all $s \in S : v_1(s) \in \{\mathbf{t}, \mathbf{f}\} \implies v_1(s) = v_2(s)$ and define for any $v \in \mathcal{V}_3$ the set of two-valued completions of it as $[v]_2 = \{w \in \mathcal{V}_2 \mid v \leq_i w\}$. For an ADF $D = (S, \Phi)$ we define the associated $\Gamma_D : \mathcal{V}_3^D \to \mathcal{V}_3^D$ as

$$\Gamma_D(v) : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\} \text{ with } s \mapsto \sqcap_i \{w(\varphi_s) \mid w \in [v]_2\}.$$

This means, the Γ-operator takes as an input a three-valued interpretation v and returns a new one $\Gamma_D(v)$. This interpretation assigns $\mathbf{t}$ ($\mathbf{f}$) to a statement s if all two-valued completion of it assigns $\mathbf{t}$ ($\mathbf{f}$) to ϕ_s. If two completions disagree on ϕ_s, s is rendered $\mathbf{u}$.

[1] If $par(s) = \emptyset$, we use the 0-ary connectives "$\top$" or "$\bot$".

Example 2 (Example 1 cont.). *Consider the three-valued interpretation* $v = \{j \mapsto \mathbf{t}, h \mapsto \mathbf{u}, m \mapsto \mathbf{u}, w \mapsto \mathbf{f}\}$. *In order to obtain the new interpretation* $\Gamma_D(v)$ *we have to compute the value* $\Gamma_D(v)[s]$ *for each single statement* $s \in S$.

1. *There are four two-valued completions* v_1, v_2, v_3, v_4 *of* v *as the statements* h *and* m *are set to* $\mathbf{u}$.
2. *We have to evaluate all acceptance formulae* ϕ *w.r.t. each single completion (depicted in Table 2).*

	j	h	m	w	ϕ_j	ϕ_h	ϕ_m	ϕ_w
v_1	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{f}$
v_2	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{f}$
v_3	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$
v_4	$\mathbf{t}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{t}$	$\mathbf{f}$

Table 2: Evaluating two-valued completions

3. *Finally, we have to build the consensus* $\sqcap_i$ *of each column in order to obtain* $\Gamma_D(v)[s]$. *Hence,* $\Gamma_D(v) = \{j \mapsto \mathbf{u}, h \mapsto \mathbf{u}, m \mapsto \mathbf{t}, w \mapsto \mathbf{f}\}$.

Now, we are ready to define three well-known semantics [Brewka *et al.*, 2013], namely admissible, complete and preferred semantics (abbr. as adm, cmp and prf).

Definition 1. *Given an ADF* $D = (S, \Phi)$ *and* $v \in \mathcal{V}_3^D$.

1. $v \in adm(D)$ *iff* $v \leq_i \Gamma_D(v)$,
2. $v \in cmp(D)$ *iff* $v = \Gamma_D(v)$ *and*
3. $v \in prf(D)$ *iff* v *is* $\leq_i$*-maximal in* $cmp(D)$.

Example 3 (Example 2 cont.). *Note that* $v \not\leq_i \Gamma_D(v)$ *since* $v(j) = \mathbf{t} \neq \mathbf{u} = \Gamma_D(v)[j]$. *Consequently,* $v \notin adm(D)$ *and therefore,* $v \notin \tau(D)$ *for* $\tau \in \{cmp, prf\}$. *We leave it to the reader to verify that* $w = \{j \mapsto \mathbf{t}, h \mapsto \mathbf{u}, m \mapsto \mathbf{t}, w \mapsto \mathbf{f}\}$ *is a complete (but not preferred) interpretation of* D.

3 Logics for ADFs: State of the Art

As we have seen in the background section the computation of the semantics of a given ADF $D = (S, \Phi)$ heavily relies on the Γ-operator which requires the consideration of all two-valued completions of a considered three-valued interpretation v. More precisely, for a specific statement s, we have $\Gamma_D(v)[s] = \sqcap_i\{w(\varphi_s) \mid w \in [v]_2\}$. As this equality does not rely on the semantics of a certain three-valued logic $\mathcal{L}_3$ the question arose from the start whether there is such a logic coinciding with the consensus operator, i.e.

> Q: Is there a three-valued logic $\mathcal{L}_3$, s.t. for any formula $\phi \in \mathcal{F}$ and three-valued interpretation $v \in \mathcal{V}_3$: $v^{\mathcal{L}_3}(\varphi) = \sqcap_i\{w(\varphi) \mid w \in [v]_2\}$?

It turned out that the answer depends on the property of truth-functionality.[2] Requiring truth-functionality yields a general negative result [Baumann and Heinrich, 2020, Theorem 1]. Dropping this property allows to find a coinciding three-valued logic, namely the so-called *possibilistic logic* [Heyninck *et al.*, 2022, Lemma 2]. Although this is a remarkable theoretical result there is still one main drawback from a conceptional as well as computational point of view, namely that the semantics of possibilistic logic is based on the so-called necessity and possibility measures which still require the consideration of two-valued completions (cf. [Heyninck *et al.*, 2022, Definition 1] or [Dubois and Prade, 1998] for more details.).

[2] Roughly speaking, a logic is truth-functional if the evaluation of a composed formulae depends of the truth values of its constituting subformulae only.

Since we are faced with the general impossibility result regarding question Q if requiring truth-functional three-valued logics, one further natural question is to ask for positive results for (interesting) syntactical subclasses $\mathcal{G} \subseteq \mathcal{F}$. So far only little is known. For instance, AF-like acceptance formulae, i.e. formulae which are conjunctions of negated atoms are covered by $\mathcal{K}_3$ [Baumann and Heinrich, 2020, Theorem 2] and [Ciucci *et al.*, 2014, Proposition 4.5]. However, the question Q is still unsolved for the prominent subclass of so-called *bipolar ADFs* representing a proper and quite expressive generalization of AF-like ADFs. Showing that also bipolar ADFs are covered by Kleene's $\mathcal{K}_3$ is the main aim of the subsequent section.

4 Bipolar ADFs - The Expressive Subclass

The class of Bipolar ADFs (BADF) has been already introduced in the first paper on ADFs [Brewka and Woltran, 2010]. It gained much attention as it was shown to be a rather expressive subclass of ADFs with attractive computational properties [Straß, 2013; 2015]. More precisely, BADFs are strictly more expressive than AFs as they provide various notions of attack and support, instead of single attacks only. Moreover, and this is the decisive point from a KR perspective, the computation with all the standard semantics is of lower computational complexity in comparison to general ADFs, but without any increase when compared to AFs [Straß and Wallner, 2015; Brewka *et al.*, 2017b].

4.1 Semantical and Syntactical Bipolarity

Let us start with the definition of BADFs. An ADF $D = (S, L, C)$ is bipolar if each of its acceptance functions C_s is *semantically bipolar* [Straß, 2015]. The latter is fulfilled if each statement is supporting, attacking or both for C_s. More precisely, given a Boolean function $f : 2^A \to \{\mathbf{t}, \mathbf{f}\}$, then

- $a \in A$ is *supporting* iff for all $M \subseteq A, f(M) = \mathbf{t}$ implies $f(M \cup \{a\}) = \mathbf{t}$,
- $a \in A$ is *attacking* iff for all $M \subseteq A, f(M) = \mathbf{f}$ implies $f(M \cup \{a\}) = \mathbf{f}$.

We write $a \in sup(f)$ or $a \in att(f)$, respectively. Note that in case of BADFs there is no $a \in A$, s.t. $a \notin sup(f)$ and $a \notin att(f)$. In [Straß, 2015] the concept of semantically bipolar functions has been linked with the sub-class of syntactically bipolar formulae. In order to define this subclass we have to recall the concept of *polarity*. An occurrence of an atom a in a formula ϕ has positive/negative polarity if the number of negations on the path from the root of the formula tree to the atom is even/odd. We define the polarity of a in formula ϕ

as:[3]

$$pol(a,\phi) = \begin{cases} \emptyset, & a \notin \sigma(\phi) \\ \{+\}, & \text{only positive occurrence of } a \text{ in } \phi \\ \{-\}, & \text{only negative occurrence of } a \text{ in } \phi \\ \{\pm\}, & \text{pos. and neg. occurrences of } a \text{ in } \phi \end{cases}$$

Now, a propositional formula ϕ is called *syntactically bipolar* if and only if no atom occurs both positively and negatively. More formally, there is no $a \in \sigma(\phi)$, s.t. $pol(a,\phi) = \{\pm\}$. The formulae $\phi = \neg((\neg a \vee b) \wedge \neg\neg b)$ and $\psi = ((a \vee b) \wedge c) \vee \neg\neg b$ are examples of syntactically bipolar formulae. We further say that two formulae have the same polarity, if they agree on the polarities of their shared atoms. This means, we (slightly abuse notation and) write $pol(\phi) = pol(\psi)$ if for any $a \in \sigma(\phi) \cap \sigma(\psi)$, $pol(a,\phi) = pol(a,\psi)$. Note that the above presented formulae do not possess the same polarity as $pol(b,\phi) = \{-\} \neq \{+\} = pol(b,\psi)$.

4.2 Representing BADFs - A Normal Form

In this section we introduce a new normal form, so-called *bipolar disjunctive normal form* (for short, *bipolar DNF)*, and show that it is expressive enough to cover semantically bipolar functions. As the name suggests we want to consider formulae which are in disjunctive normal form and are also bipolar. In order to prove our main result we provide an inductive definition enabling structural induction. As building blocks we use the set $\mathcal{C}$ containing cubes over A which does not allow multiple occurrences of atoms. Remember that a cube is a conjunction of literals. This means, $\phi = a \wedge b \wedge \neg c$ is in $\mathcal{C}$ whereas neither $\psi = (a \wedge b) \vee c$ (no cube), nor $\xi = a \wedge b \wedge \neg a$ (multiple occurrences of a) possess this property. Now we are ready to define formulae in bipolar DNF inductively.

Definition 2. *The set $\mathcal{F}_b$ of bipolar DNFs is defined as:*

1. $\mathcal{C} \subseteq \mathcal{F}_b$,
2. *If $\phi, \psi \in \mathcal{F}_b$ and $pol(\phi) = pol(\psi)$, then $\phi \vee \psi \in \mathcal{F}_b$.*

Note that the condition $pol(\phi) = pol(\psi)$ enforces that different occurrences of an atom agree on their polarity. This means, the formula $(a \wedge b \wedge \neg c) \vee (b \wedge \neg d)$ is a bipolar DNF and $(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg d)$ is not. We now formally prove that bipolar disjunctive normal forms are indeed syntactically bipolar as their name suggests.

Proposition 1. *Bipolar DNFs are syntactically bipolar.*

Proof. We prove the assertion by structural induction.

- (base case) Let $\varphi \in \mathcal{C}$. By definition, we have that any atom occurs at most once in φ. Hence, there is no $a \in \sigma(\phi)$, s.t. $pol(a,\phi) = \{\pm\}$.
- (induction step) Given $\phi, \psi \in \mathcal{F}_b$ with $pol(\phi) = pol(\psi)$ and ϕ as well as ψ are syntactically bipolar. Consider $\phi \vee \psi$. Towards a contradiction assume that $\phi \vee \psi$ is not syntactically bipolar. This means, there is an $a \in \sigma(\phi \vee \psi)$, s.t. $pol(a, \phi \vee \psi) = \{\pm\}$. Note that $\sigma(\phi \vee \psi) = \sigma(\phi) \cup \sigma(\psi)$. Case distinction:
  1. $a \in \sigma(\phi) \setminus \sigma(\psi)$. Impossible as ϕ is assumed to be syntactically bipolar.
  2. $a \in \sigma(\phi) \cap \sigma(\psi)$. Impossible as $pol(\phi) = pol(\psi)$ guarantees that for any $a \in \sigma(\phi) \cap \sigma(\psi)$, $pol(a,\phi) = pol(a,\psi)$.
  3. $a \in \sigma(\psi) \setminus \sigma(\phi)$. Impossible as ψ is assumed to be syntactically bipolar. □

[3]The set $\sigma(\phi)$ contains all atoms occurring in ϕ.

We now turn to the main result of this section allowing us to link the class of bipolar ADFs with bipolar DNFs. More precisely, we show that semantically bipolar functions can be represented as bipolar DNFs. Moreover, any bipolar DNF canonically induces a semantically bipolar function. The result is a strengthening of a former correspondence result [Straß, 2015, Theorem 1] and will be the decisive ingredient to show that BADFs are covered by Kleene's $\mathcal{K}_3$.

Theorem 1.

1. *For each formula ϕ in bipolar DNF we have that*

$$f_\phi : 2^{\sigma(\phi)} \to \{\mathbf{t},\mathbf{f}\} \text{ with } M \mapsto v_M(\phi)$$

is semantically bipolar. The interpretation v_M is obtained from M via: For each $a \in \sigma(\phi)$, $v_M(a) = \mathbf{t}$ iff $a \in M$.

2. *For each semantically bipolar function $f : 2^A \to \{\mathbf{t},\mathbf{f}\}$, a bipolar DNF ψ_f with $f_{\psi_f} = f$ is given by:*

$$\psi_f = \bigvee_{\substack{M \subseteq A,\\ f(M)=\mathbf{t}}} \psi_M \quad \text{with} \quad \psi_M = \bigwedge_{\substack{a \in M,\\ a \notin att(f)}} a \wedge \bigwedge_{\substack{a \in A \setminus M,\\ a \notin sup(f)}} \neg a$$

Proof.

1. Given a formula ϕ in bipolar DNF. In Proposition 1 we have shown that formulae in bipolar DNF are syntactically bipolar. Hence, we me apply [Straß, 2015, Item 1 of Theorem 1] and obtain the semantical bipolarity of the presented function.
2. The construction of ψ_f was already given in [Straß, 2015, Item 2 of Theorem 1]. It remains to check that this formula is indeed in bipolar disjunctive normal form. First, for each set M, the associated formula ψ_M is a cube without multiple occurrences of atoms as $a \in M$ and $a \in A \setminus M$ is impossible. This means, for each M, $\psi_M \in \mathcal{C}$. Moreover, since by definition a semantically bipolar function possesses arguments which are supporting, attacking or both, i.e. $a \notin sup(f)$ and $a \notin att(f)$ is impossible, we deduce the impossibility of having $pol(a,\psi_M) = \{+\}$ and $pol(a,\psi_{M'}) = \{-\}$ for two cubes ψ_M and $\psi_{M'}$. Hence, ψ_f meets any criteria of a bipolar DNF. □

To get an idea of the constructions used in Theorem 1 we present the following example.

Example 4. *Consider the bipolar DNF $\phi = a \vee \neg b$. We obtain $f_\phi = \{\emptyset \mapsto \mathbf{t}, \{a\} \mapsto \mathbf{t}, \{b\} \mapsto \mathbf{f}, \{a,b\} \mapsto \mathbf{t}\}$. In accordance with Item 1 f_ϕ is semantically bipolar since $a \in sup(f_\phi)$ and $b \in att(f_\phi)$. Moreover, f_ϕ induces the bipolar DNF $\psi_{f_\phi} = \psi_\emptyset \vee \psi_{\{a\}} \vee \psi_{\{a,b\}}$ with $\psi_\emptyset = \neg b$, $\psi_{\{a\}} = a \wedge \neg b$ and $\psi_{\{a,b\}} = a$. Since ψ_{f_ϕ} and ϕ possess the same two-valued models we verify $f_{\psi_{f_\phi}} = f_\phi$ as claimed in Item 2.*

4.3 Kleene's $\mathcal{K}_3$ Covers BADFs

We are now prepared for the main result of this paper. That is, we show that Kleene's three-valued logic serves for bipolar DNFs which themselves correspond to the class of bipolar ADFs. More precisely, analogously to AF-like acceptance formulae, it is possible to evaluate a bipolar DNF φ w.r.t. $\mathcal{K}_3$ instead of computing all two-valued completions and applying the consensus operator afterwards as required in the original definition.

Theorem 2. *For any $\varphi \in \mathcal{F}_b$ and any $v \in \mathcal{V}_3$:*

$$v^{\mathcal{K}_3}(\varphi) = \sqcap_i \{w(\varphi) \mid w \in [v]_2\}.$$

Proof. We prove the assertion by structural induction.

- (base case) Let $\varphi \in \mathcal{C}$. This means, φ is of the form $a_1 \wedge \ldots \wedge a_n \wedge \neg b_1 \wedge \ldots \wedge \neg b_m$ with $a_i, b_j \in A$ and some $n, m \in \mathbb{N}$. Note that one of the two might be zero. Given $v \in \mathcal{V}_3$. Case distinction.
  1. Let $v^{\mathcal{K}_3}(\varphi) = \mathbf{t}$. Hence, according to Table 1 we have that each single conjunct of ϕ has to be evaluated to true, i.e. $v^{\mathcal{K}_3}(a_i) = \mathbf{t}$ and $v^{\mathcal{K}_3}(\neg b_j) = \mathbf{t}$. The latter implies $v^{\mathcal{K}_3}(b_j) = \mathbf{f}$. Consequently, since v is already two-valued, there are no further two-valued completions, i.e. $[v]_2 = \{v\}$. Since Kleene's $\mathcal{K}_3$ coincide with propositional logic in the two-valued case, i.e. $v(\varphi) = \mathbf{t}$, we deduce $\sqcap_i\{w(\varphi) \mid w \in [v]_2\} = \mathbf{t}$.
  2. Let $v^{\mathcal{K}_3}(\varphi) = \mathbf{f}$. Thus, according to Kleene at least one conjunct has to evaluated to false. This means, there is one a_i with $v(a_i) = \mathbf{f}$ or alternatively, one b_j with $v(b_j) = \mathbf{t}$. Now, no matter which two-valued completion $w \in [v]_2$ is considered, w and v agree on the evaluation of this atom implying that $w(\phi) = \mathbf{f}$ even in the propositional case. Consequently, $\sqcap_i\{w(\varphi) \mid w \in [v]_2\} = \sqcap_i\{\mathbf{f}\} = \mathbf{f}$.
  3. Let $v^{\mathcal{K}_3}(\varphi) = \mathbf{u}$. Hence, according to Table 1 at least one conjunct has to evaluated to unknown (1) and no conjunct evaluates to false (2).
     Condition (1) implies $v^{\mathcal{K}_3}(a_i) = \mathbf{u}$ or $v^{\mathcal{K}_3}(\neg b_j) = \mathbf{u} = v^{\mathcal{K}_3}(b_j)$ for some a_i or b_j. This insight allows to render the conjunction false w.r.t. a two-valued completion $w \in [v]_2$ via $w(a_i) = \mathbf{f}$ or $w(b_j) = \mathbf{t}$, respectively. Note that the assignment for the remaining atoms does not matter as a conjunction evaluates to false, whenever at least one conjunct is false.
     The second condition (2) implies that $w \in \mathcal{V}_2$ with $w = \{a_1, \ldots, a_n \mapsto \mathbf{t}, b_1, \ldots, b_n \mapsto \mathbf{f}\}$ is a two-valued completion of v, i.e. $w \in [v]_2$. By construction $w(\phi) = \mathbf{t}$. Consequently, $\sqcap_i\{w(\varphi) \mid w \in [v]_2\} = \sqcap_i\{\mathbf{t}, \mathbf{f}\} = \mathbf{u}$ concluding the last case.
- (induction step) Given $\phi, \psi \in \mathcal{F}_b$ with $pol(\phi) = pol(\psi)$ as well as $v^{\mathcal{K}_3}(\varphi) = \sqcap_i\{w(\varphi) \mid w \in [v]_2\}$ and $v^{\mathcal{K}_3}(\psi) = \sqcap_i\{w(\psi) \mid w \in [v]_2\}$ for any three-valued interpretation v. We have to show that this conveys to the disjunction, i.e. $v^{\mathcal{K}_3}(\varphi \vee \psi) = \sqcap_i\{w(\varphi \vee \psi) \mid w \in [v]_2\}$ for any three-valued interpretation v. Case distinction.
  1. Let $v^{\mathcal{K}_3}(\varphi) = \mathbf{t}$ or $v^{\mathcal{K}_3}(\psi) = \mathbf{t}$. (at least one $\mathbf{t}$)
     Hence, according to Table 1 we obtain $v^{\mathcal{K}_3}(\varphi \vee \psi) = \mathbf{t}$. Without loss of generality assume $v^{\mathcal{K}_3}(\varphi) = \mathbf{t}$. By induction hypothesis we obtain $\mathbf{t} = \sqcap_i\{w(\varphi) \mid w \in [v]_2\}$ yielding $w(\varphi) = \mathbf{t}$ for any $w \in [v]_2$. Consequently, $w(\varphi \vee \psi) = \mathbf{t}$ for any $w \in [v]_2$ proving $\mathbf{t} = \sqcap_i\{w(\varphi \vee \psi) \mid w \in [v]_2\}$.
  2. Let $v^{\mathcal{K}_3}(\varphi) = v^{\mathcal{K}_3}(\psi) = \mathbf{f}$. (both $\mathbf{f}$)
     Hence, $v^{\mathcal{K}_3}(\varphi \vee \psi) = \mathbf{f}$ according to Table 1. By induction hypothesis we obtain $\sqcap_i\{w(\varphi) \mid w \in [v]_2\} = \mathbf{f} = \sqcap_i\{w(\psi) \mid w \in [v]_2\}$ yielding $w(\varphi) = \mathbf{f}$ for any $w \in [v]_2$. Consequently, $w(\varphi \vee \psi) = \mathbf{f}$ for any $w \in [v]_2$ proving $\mathbf{f} = \sqcap_i\{w(\varphi \vee \psi) \mid w \in [v]_2\}$.
  3. Let $v^{\mathcal{K}_3}(\varphi) = v^{\mathcal{K}_3}(\psi) = \mathbf{u}$. (both $\mathbf{u}$)
     Consequently, $v^{\mathcal{K}_3}(\varphi \vee \psi) = \mathbf{u}$ according to Table 1. Applying induction hypothesis justifies $\sqcap_i\{w(\varphi) \mid w \in [v]_2\} = \mathbf{u} = \sqcap_i\{w(\psi) \mid w \in [v]_2\}$. This means, there are at least two completions $w^{\phi}_{\mathbf{t}}, w^{\phi}_{\mathbf{f}} \in [v]_2$, s.t. $w^{\phi}_{\mathbf{t}}(\varphi) = \mathbf{t}$ and $w^{\phi}_{\mathbf{f}}(\varphi) = \mathbf{f}$. The same applies to ψ. This means, there are two completions $w^{\psi}_{\mathbf{t}}, w^{\psi}_{\mathbf{f}} \in [v]_2$, s.t. $w^{\psi}_{\mathbf{t}}(\psi) = \mathbf{t}$ and $w^{\psi}_{\mathbf{f}}(\psi) = \mathbf{f}$. Now, we have to show that there are two completions $w^{\phi\vee\psi}_{\mathbf{t}}, w^{\phi\vee\psi}_{\mathbf{f}} \in [v]_2$, s.t. $w^{\phi\vee\psi}_{\mathbf{t}}(\phi \vee \psi) = \mathbf{t}$ (1) and $w^{\phi\vee\psi}_{\mathbf{f}}(\phi \vee \psi) = \mathbf{f}$ (2). This existence guarantees $\mathbf{u} = \sqcap_i\{w(\varphi \vee \psi) \mid w \in [v]_2\}$ as required.
     (1) Consider $w^{\phi\vee\psi}_{\mathbf{t}} = w^{\psi}_{\mathbf{t}}$. Since $w^{\phi}_{\mathbf{t}}(\varphi) = \mathbf{t}$ is assumed, we immediately obtain $w^{\phi}_{\mathbf{t}}(\varphi \vee \psi) = \mathbf{t}$ and we are done.
     (2) Since $\phi, \psi \in \mathcal{F}_b$ we may deduce that
     $$\phi = \bigvee_{i=1}^{n} P_i \text{ and } \psi = \bigvee_{i=1}^{m} Q_i \text{ with } P_i, Q_i \in \mathcal{C}.$$
     Since $w^{\phi}_{\mathbf{f}}(\varphi) = \mathbf{f}$ we infer $w^{\phi}_{\mathbf{f}}(P_i) = \mathbf{f}$ for any $1 \leq i \leq n$. Since any P_i is a conjunction of literals, there has to be at least one atom $p_i \in \sigma(P_i)$, s.t. $w^{\phi}_{\mathbf{f}}(p_i) = \mathbf{f}$ if $pol(p_i, P_i) = \{+\}$ and $w^{\phi}_{\mathbf{f}}(p_i) = \mathbf{t}$ if $pol(p_i, P_i) = \{-\}$. For convenience, let us call such an atom p_i *falsifier* of P_i. Since $pol(\phi) = pol(\psi)$ is assumed we have that any falsifier p_i of a certain P_i is also a falsifier of Q_i, whenever $p_i \in \sigma(Q_i)$. Let us pick one single falsifier p_i for each P_i regarding $w^{\phi}_{\mathbf{f}}$. This returns a set $P = \{p_1, \ldots, p_n\}$. Now, choose for the remaining Q_i which are not already falsified by some $p \in P$ under $w^{\phi}_{\mathbf{f}}$ a single falsifier q_i regarding $w^{\psi}_{\mathbf{f}}$. Let us denote this set with $Q = \{q_1, ..., q_s\}$. Note that $s = 0$, i.e. $Q = \emptyset$ is possible. Moreover, P and Q are disjoint by definition. Now we are ready to define a falsifying interpretation $w^{\phi\vee\psi}_{\mathbf{f}}$, namely
     $$w^{\phi\vee\psi}_{\mathbf{f}}(a) = \begin{cases} w^{\phi}_{\mathbf{f}}(a), & \text{if } a \in P \\ w^{\psi}_{\mathbf{f}}(a), & \text{if } a \in Q \\ 0, & \text{otherwise} \end{cases}$$

Consequently, $w_{\mathbf{f}}^{\phi\vee\psi}(\phi\vee\psi) = \mathbf{f}$ by construction.

4. Let $v^{\mathcal{K}_3}(\varphi) = \mathbf{u}$ and $v^{\mathcal{K}_3}(\psi) = \mathbf{f}$. (one $\mathbf{u}$, one $\mathbf{f}$) Consequently, $v^{\mathcal{K}_3}(\varphi \vee \psi) = \mathbf{u}$. By induction hypothesis, there are at least two completions $w_{\mathbf{t}}^{\phi}, w_{\mathbf{f}}^{\phi} \in [v]_2$, s.t. $w_{\mathbf{t}}^{\phi}(\varphi) = \mathbf{t}$ and $w_{\mathbf{f}}^{\phi}(\varphi) = \mathbf{f}$. Moreover, $w(\psi) = \mathbf{f}$ for any $w \in [v]_2$. Hence, we have already found two witnessing completions since $w_{\mathbf{t}}^{\phi}(\varphi \vee \psi) = \mathbf{t}$ and $w_{\mathbf{f}}^{\phi}(\varphi \vee \psi) = \mathbf{f}$ according to Table 1. Thus, we have shown $\sqcap_i\{w(\varphi\vee\psi) \mid w \in [v]_2\} = \sqcap_i\{\mathbf{t},\mathbf{f}\} = \mathbf{u}$. □

Example 5 (Revisiting Example 2)**.** *Consider again the three-valued interpretation $v = \{j \mapsto \mathbf{t}, h \mapsto \mathbf{u}, m \mapsto \mathbf{u}, w \mapsto \mathbf{f}\}$. We observe that any non-trivial acceptance condition is in bipolar DNF. More precisely, $\phi_j = h \vee (m \wedge \neg w)$ and $\phi_h = h$ are bipolar DNFs. Now, according to the main theorem we may simply apply Kleene's $\mathcal{K}_3$ instead of computing two-valued completions. More precisely, $\Gamma_D(v)[j] = v^{\mathcal{K}_3}(\phi_j) = v^{\mathcal{K}_3}((h \vee m) \wedge \neg w) = \mathbf{u}$ coinciding with $\sqcap_i\{w(\phi_j) \mid w \in [v]_2\}$. The latter was calculated with enormous effort in Example 2.*

4.4 Beyond Bipolar DNFs and Representational Issues

Before closing the main section and turning to experimental results we want to comment on semantically equivalent replacements. Remember that two formulae are said to be semantically equivalent if and only if both have same truth value under all interpretations. Regarding replacements, one has to be aware of the fact that semantical equivalence w.r.t. Kleene's $\mathcal{K}_3$ (for short, $\equiv_{\mathcal{K}_3}$) and propositional logic differ. More precisely, the former implies the latter but not necessarily vice versa. For instance, consider $\phi = a$ and $\psi = a \vee (b \wedge \neg b)$. Both are considered as semantically equivalent in propositional logic, i.e. $w(\phi) = w(\psi)$ for each $w \in \mathcal{V}_2$. However, the three-valued interpretation $v = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}\}$ witnesses their non-equivalence w.r.t. $\mathcal{K}_3$ as $v_3^{\mathcal{K}}(\phi) = \mathbf{f} \neq \mathbf{u} = v_3^{\mathcal{K}}(\psi)$. This means, applying semantically equivalent replacements to an acceptance formula in bipolar DNF does not guarantee the applicability of Theorem 2 if done w.r.t. propositional logic. In contrast replacing bipolar DNFs with formulae semantically equivalent w.r.t. Kleene yields a class beyond bipolar DNFs also covered by the main result. For instance, although $\xi = \neg(a \wedge b)$ is obviously not in bipolar DNF it is covered by Theorem 2 as $v_3^{\mathcal{K}}(\xi) = v_3^{\mathcal{K}}(\neg a \vee \neg b)$ for each $v \in \mathcal{V}_3$.

The following result shows that any syntactically bipolar formula (like ξ) is semantically equivalent in terms of $\mathcal{K}_3$ to a bipolar DNF. This result justifies the name bipolar disjunctive normal form even in the realm of Kleene's three-valued logic.

Proposition 2. *For any syntactically bipolar formula ϕ, there exists a formula ψ in bipolar DNF, s.t. $\phi \equiv_{\mathcal{K}_3} \psi$.*

Proof (sketch). The following semantical equivalences are known from classical logic, but also hold for Kleene's $\mathcal{K}_3$ (cf. [Hölldobler and Kencana Ramli, 2009, Table 2]).

1. Use *Double Negation* and *De Morgan* to reach the so-called negation normal form, i.e. negations are only in front of atoms.
2. Use *Distributivity, Commutativity* and *Associativity* to obtain a classical DNF.
3. Apply *Idempotency* to guarantee that each disjunct is in $\mathcal{C}$, i.e. a cube without multiple occurrences of atoms.

It remains to argue that the obtained DNF is indeed bipolar. This property follows from the fact that each single replacement does not change the polarity of the atoms. □

The former representation result allows us to extend the main theorem to the whole class of syntactically bipolar formulae. This result frees us from the strict syntactic corset of bipolar DNFs.

Theorem 3. *For any syntactically bipolar formula ϕ and any three-valued interpretation $v \in \mathcal{V}_3$:*

$$v^{\mathcal{K}_3}(\varphi) = \sqcap_i\{w(\varphi) \mid w \in [v]_2\}.$$

Proof. Combine Proposition 2 and Theorem 2. □

5 Complexity and Experimental Results

The complexity of the standard decision problems like *Verification*, *Credulous Acceptance* or *Sceptical Acceptance* highly depend on the complexity of the Γ-operator [Straß and Wallner, 2015; Dvorák and Dunne, 2017]. For instance, verifying that a given three-valued interpretation v is admissible in an ADF D requires to check whether $v \leq_i \Gamma_D(v)$. This means, we have to examine $v(s) \leq_i \Gamma_D(v)[s] = \sqcap_i\{w(\varphi_s) \mid w \in [v]_2\}$ for each single statement s. Consequently, if for instance $v(s) = \mathbf{t}$, we have to verify that $\Gamma_D(v)[s] = \mathbf{t}$. This means, φ_s has to evaluate to true for each single $w \in [v]_2$. Clearly, this reminds of the validity problem in propositional logic. And indeed, it was shown that verifying admissibility for general ADFs is coNP-complete [Wallner, 2014]. Let us now assume that we are faced with a syntactically bipolar acceptance formula ϕ_s. In this case verifying $v(s) \leq_i \Gamma_D(v)[s]$ becomes tractable as it suffices to evaluate φ_s w.r.t. $\mathcal{K}_3$ and to check whether $v(s) \leq_i v^{\mathcal{K}_3}(\varphi_s)$. This means, applying Theorem 3 yields a P algorithm for verifying admissibility in case of BADFs. Again, this membership result is not new [Straß and Wallner, 2015], but the way to achieve it is. In the following we will show how we may computationally benefit from applying the main theorem.

For this very first experiment, we considered bipolar ADFs with a number of statements between 1 and 12. For each number, we generated at least 100 test instances, i.e. 100 ADFs with different bipolar acceptance formulae. The tests were run on an Ubuntu desktop with an Intel i5-6400 CPU and 32 GiB RAM. The implemented Python script systematically generates and checks all three-valued interpretations v according to the specific semantical requirements, e.g. $v \leq_i \Gamma_D(v)$ in the case of admissible semantics. For a particular statement s, the script calculates $\Gamma_D(v)[s]$ in two different ways: via the classical consensus, i.e. with the help of two-valued completions, and via applying Kleene's three-valued logic. We measured the time required for each type of calculation, considering a 30 minute limit.

The results for admissible, complete and preferred semantics are depicted in Table 3. The table is structured as follows: First, column "n" indicates the number of statements. Secondly, "σ-two" and "σ-tri" announce how σ-interpretations are obtained, i.e. via two-valued completions or three-valued logic, respectively. Finally, the factor in column "speed" indicates how much faster the three-valued approach is compared to the two-valued one. Results are rounded to four digits after the decimal point, enabling runtimes like 0.0 as well as matching entries. The entry "time out" indicates that the calculation was stopped as the 30 minute limit was reached. We mention that even for the Kleene method, statement numbers over 12 timed out.

n	adm-two	adm-tri	Speed	cmp-two	cmp-tri	Speed	prf-two	prf-tri	Speed
1	0.0001	0.0001	0.93	0.0001	0.0001	0.92	0.0001	0.0001	0.93
2	0.0003	0.0003	1.04	0.0003	0.0003	1.04	0.0004	0.0003	1.04
3	0.0021	0.0017	1.22	0.0021	0.0018	1.22	0.0021	0.0018	1.22
4	0.0126	0.0083	1.51	0.0127	0.0084	1.51	0.0126	0.0084	1.51
5	0.0842	0.0408	2.06	0.0844	0.041	2.06	0.0846	0.0408	2.08
6	0.4207	0.1765	2.38	0.4323	0.1765	2.45	0.4288	0.1765	2.43
7	2.6096	0.6686	3.9	2.608	0.6706	3.89	2.6064	0.6671	3.91
8	14.2955	2.6573	5.38	14.2891	2.6586	5.37	14.275	2.6523	5.38
9	81.3765	11.0362	7.37	81.3055	11.0185	7.38	81.2916	11.0362	7.37
10	421.8405	42.2137	9.99	421.6861	42.2658	9.98	421.3962	42.1764	9.99
11	time out	168.0131	–	time out	168.2505	–	time out	168.156	–
12	time out	615.5548	–	time out	615.5211	–	time out	616.5835	–

Table 3: Runtimes for admissible, complete and preferred semantics

The main aim was to show that the use of Kleene's three-valued logic is significantly faster than the classical approach using two-valued completions. Indeed, the runtimes depicted in Table 3 show this impressively. However, both methods show exponential runtime growth, yielding a linear function on a logarithmic scale (cf. Figure 2). This is not surprising since the number of interpretations to be tested also depends exponentially on the number of statements. It can be observed that there are no significant differences in the runtimes among the considered semantics. According to the solver design the computation of complete/preferred interpretations should be slightly slower than that of admissible/complete interpretations as the former require the latter. Since the number of admissible/complete interpretations used to filter out complete/preferred interpretations is rather low, the additional time for comparing these interpretation is negligibly small and not visible in the results. In order to see these differences more clearly, a larger number of statements, test instances, and measurements are required. To ensure reproducability, we provide access to our repository[4], containing the solver, test cases, and other relevant technical details.

6 Discussion and Conclusion

The topic of finding underlying logics for ADFs is not new. To the best of our knowledge the first result was given by Bochman who showed that there is a uniform and modular translation from ADFs into causal reasoning. For each acceptance function ϕ_s two causal rules are introduced, namely $\phi_s \Rightarrow s$ and $\neg\phi_s \Rightarrow \neg s$. It is then shown that the obtained causal theory, or more precisely, its induced causal operator relates semantically to the Γ-operator in a way which allows to show a broad correspondence between the two formalisms [Bochman, 2016]. A correspondence result without translations was later given by Heyninck and colleagues. They showed that the three-valued possibilistic logic coincides with the results of the Γ-operator [Heyninck *et al.*, 2022]. This logic is not truth-functional which is indeed unavoidable due to an impossibility result in [Baumann and Heinrich, 2020].

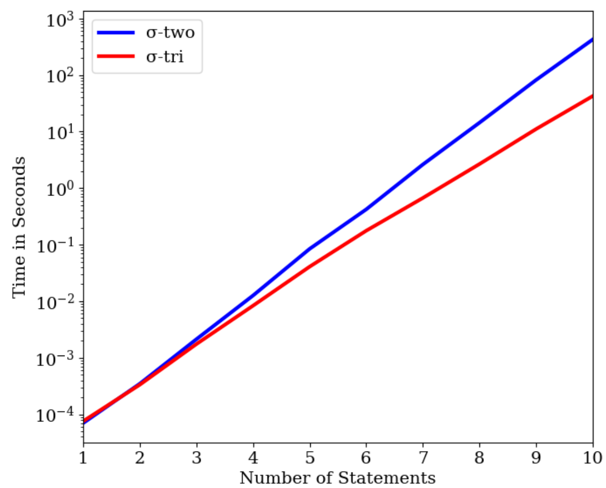

Figure 2: Performance illustration on logarithmic scale

In this paper we proved that in case of BADFs we may use Kleene's $\mathcal{K}_3$ as an underlying truth-functional logic. This logic allows to bypass the computationally intensive procedure of considering all two-valued completions which was still implicit in possibilistic logic. To prove the main correspondence theorem, we introduced the so-called bipolar DNFs inspired by previous results in [Straß, 2015]. The decisive point of the proof was to construct a verifying or falsifying interpretation for a compound formula given verifying or falsifying interpretations of its subformulae. Bipolar DNFs proved to be extremely helpful in this regard as their structure give us sufficient control over the considered interpretations. Finally, we free us from the syntactic restrictions of bipolar DNFs and extended the correspondence result to syntactically bipolar formulae in general.

The achieved correspondence result is in the first place a theoretical one, which clarifies the open question whether some and which three-valued logic underlies BADFs. In the future, this main result can be used for both, further theoretical issues like (re)considering complexity questions as well as practical applications such as developing new algorithms relying on Kleene's $\mathcal{K}_3$. One promising idea is to combine these newly developed computing methods with already existing procedures. For instance, one may think of a preprocessing step checking which acceptance formulae are syntactically bipolar. For the identified statements we may use the faster Kleene-procedure. We indicated the computational benefit of using Kleene with some first experiments. However, it will be interesting to see how combined methods may improve more sophisticated ADF solvers like *k++ADF* implemented in C++ programming language [Linsbichler *et al.*, 2022], *goDIAMOND* [Ellmauthaler and Strass, 2016] and *YADF* [Brewka *et al.*, 2017a] both based on answer set programming or *Tweety* using Java [Thimm, 2014; 2017].

[4] https://github.com/kmax-tech/IJCAI-23

Ethical Statement

There are no ethical issues.

Acknowledgments

This work was supported by the German Research Foundation (DFG, BA 6170/2-1 and BA 6170/3-1) and the German Federal Ministry of Education and Research (BMBF, 01/S18026A-F) by funding the competence center for Big Data and AI "ScaDS.AI" Dresden/Leipzig.

References

[Alviano *et al.*, 2016] Mario Alviano, Wolfgang Faber, and Hannes Strass. Boolean functions with ordered domains in answer set programming. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 879–885. AAAI Press, 2016.

[Amgoud and Vesic, 2011] Leila Amgoud and Srdjan Vesic. A new approach for preference-based argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, 63(2):149–183, 2011.

[Atkinson *et al.*, 2017] Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Ricardo Simari, Matthias Thimm, and Serena Villata. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, 2017.

[Baroni *et al.*, 2009] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. Encompassing attacks to attacks in abstract argumentation frameworks. In Claudio Sossai and Gaetano Chemello, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 10th European Conference, ECSQARU 2009, Verona, Italy, July 1-3, 2009. Proceedings*, volume 5590 of *Lecture Notes in Computer Science*, pages 83–94. Springer, 2009.

[Baroni *et al.*, 2018] P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre. *Handbook of Formal Argumentation*. College Publications, 2018.

[Baumann and Heinrich, 2020] Ringo Baumann and Maximilian Heinrich. Timed abstract dialectical frameworks: A simple translation-based approach. In *Computational Models of Argument - Proceedings of COMMA 2020, Perugia, Italy, September 4-11, 2020*, pages 103–110, 2020.

[Baumann and Straß, 2017] Ringo Baumann and Hannes Straß. On the number of bipolar boolean functions. *Journal of Logic and Computation*, 27(8):2431–2449, 2017.

[Bench-Capon and Atkinson, 2009] Trevor J. M. Bench-Capon and Katie Atkinson. Abstract argumentation and values. In *Argumentation in Artificial Intelligence*, pages 45–64. 2009.

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

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

[Bochman, 2016] Alexander Bochman. Abstract dialectical argumentation among close relatives. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 127–138. IOS Press, 2016.

[Brewka and Woltran, 2010] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press, 2010.

[Brewka *et al.*, 2013] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*. IJCAI/AAAI, August 2013.

[Brewka *et al.*, 2014] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29(1):30–38, 2014.

[Brewka *et al.*, 2017a] Gerhard Brewka, Martin Diller, Georg Heissenberger, Thomas Linsbichler, and Stefan Woltran. Solving advanced argumentation problems with answer-set programming. In Satinder Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1077–1083. AAAI Press, 2017.

[Brewka *et al.*, 2017b] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks. an overview. *IfCoLog Journal of Logics and their Applications*, 4(8):2263–2317, October 2017.

[Cayrol and Lagasquie-Schiex, 2009] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolar abstract argumentation systems. In *Argumentation in Artificial Intelligence*, pages 65–84. 2009.

[Ciucci *et al.*, 2014] Davide Ciucci, Didier Dubois, and Jonathan Lawry. Borderline vs. unknown: comparing three-valued representations of imperfect information. *International Journal of Approximate Reasoning*, 55(9):1866–1889, 2014.

[Cocarascu and Toni, 2016] Oana Cocarascu and Francesca Toni. Argumentation for machine learning: A survey. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 219–230. IOS Press, 2016.

[Dubois and Prade, 1998] Didier Dubois and Henri Prade. *Possibility Theory: Qualitative and Quantitative Aspects*, pages 169–226. Springer Netherlands, Dordrecht, 1998.

[Dung, 1995] Phan Minh 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.

[Dvorák and Dunne, 2017] Wolfgang Dvorák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. *FLAP*, 4(8), 2017.

[Ellmauthaler and Strass, 2016] Stefan Ellmauthaler and Hannes Strass. DIAMOND 3.0 - A native C++ implementation of DIAMOND. In *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016*, pages 471–472, 2016.

[Fitting, 1991] Melvin Fitting. Kleene's logic, generalized. *Journal of Logic and Computation*, 1(6):797–810, 1991.

[Heyninck *et al.*, 2022] Jesse Heyninck, Gabriele Kern-Isberner, Tjitze Rienstra, Kenneth Skiba, and Matthias Thimm. Possibilistic logic underlies abstract dialectical frameworks. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI'22)*, July 2022.

[Hölldobler and Kencana Ramli, 2009] Steffen Hölldobler and Carroline Dewi Puspa Kencana Ramli. Logic programs under three-valued lukasiewicz semantics. In *Logic Programming, 25th International Conference, ICLP 2009*, pages 464–478, 2009.

[Kleene, 1952] Stephen Cole Kleene. *Introduction to Metamathematics*. Princeton, NJ, USA: North Holland, 1952.

[Linsbichler *et al.*, 2022] Thomas Linsbichler, Marco Maratea, Andreas Niskanen, Johannes Peter Wallner, and Stefan Woltran. Advanced algorithms for abstract dialectical frameworks based on complexity analysis of subclasses and sat solving. *Artificial Intelligence*, 307, June 2022.

[Nielsen and Parsons, 2006] Søren Holbech Nielsen and Simon Parsons. A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *Workshop on Argumentation in Multi-Agent Systems*, pages 54–73, 2006.

[Straß and Wallner, 2015] Hannes Straß and Johannes Peter Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence*, 226:34–74, 2015.

[Straß, 2013] Hannes Straß. Approximating Operators and Semantics for Abstract Dialectical Frameworks. *Artificial Intelligence*, 205:39–70, 2013.

[Straß, 2015] Hannes Straß. Expressiveness of two-valued semantics for abstract dialectical frameworks. *Journal of Artificial Intelligence Research*, 54:193–231, 2015.

[Straß, 2018] Hannes Straß. Instantiating rule-based defeasible theories in abstract dialectical frameworks and beyond. *Journal of Logic and Computation*, 28(3):605–627, 2018.

[Thimm, 2014] Matthias Thimm. Tweety - a comprehensive collection of java libraries for logical aspects of artificial intelligence and knowledge representation. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*, July 2014.

[Thimm, 2017] Matthias Thimm. The tweety library collection for logical aspects of artificial intelligence and knowledge representation. *Künstliche Intelligenz*, 31(1):93–97, March 2017.

[Wallner, 2014] Johannes Wallner. *Complexity results and algorithms for argumentation: Dung's frameworks and beyond*. PhD thesis, Vienna University of Technology, Institute of Information Systems, 2014.