# Expressiveness of Two-Valued Semantics for Abstract Dialectical Frameworks 

Hannes Strass<br>STRASS@INFORMATIK.UNI-LEIPZIG.DE<br>Computer Science Institute, Leipzig University<br>Augustusplatz 10, 04109 Leipzig, Germany


#### Abstract

We analyse the expressiveness of Brewka and Woltran's abstract dialectical frameworks for two-valued semantics. By expressiveness we mean the ability to encode a desired set of two-valued interpretations over a given propositional vocabulary $A$ using only atoms from $A$. We also compare ADFs' expressiveness with that of (the two-valued semantics of) abstract argumentation frameworks, normal logic programs and propositional logic. While the computational complexity of the two-valued model existence problem for all these languages is (almost) the same, we show that the languages form a neat hierarchy with respect to their expressiveness. We then demonstrate that this hierarchy collapses once we allow to introduce a linear number of new vocabulary elements. We finally also analyse and compare the representational succinctness of ADFs (for two-valued model semantics), that is, their capability to represent two-valued interpretation sets in a space-efficient manner.


## 1. Introduction

More often than not, different knowledge representation languages have conceptually similar and partially overlapping intended application areas. What are we to do if faced with an application and a choice of several possible knowledge representation languages which could be used for the application? One of the first axes along which to compare different formalisms that comes to mind is computational complexity: if a language is computationally too expensive when considering the problem sizes typically encountered in practice, then this is a clear criterion for exclusion. But what if the available language candidates have the same computational complexity? If their expressiveness in the computationalcomplexity sense of "What kinds of problems can the formalism solve?" is the same, we need a more fine-grained notion of expressiveness. In this paper, we use such a notion and study the expressiveness of abstract dialectical frameworks (ADFs) (Brewka \& Woltran, 2010; Brewka, Ellmauthaler, Strass, Wallner, \& Woltran, 2013), a recent generalisation of abstract argumentation frameworks (AFs) (Dung, 1995).

Argumentation frameworks are the de-facto standard formalism in abstract argumentation, a field that studies how (abstract) arguments relate to each other in terms of directed conflicts ("attacks"), and how these conflicts can be resolved without "looking into" the arguments. While AFs are popular and well-studied, it has been noted many times in the literature that their expressive capabilities are somewhat limited. This has only recently been made technically precise by Dunne, Dvořák, Linsbichler, and Woltran (2014, 2015), who basically showed that introducing new, purely technical arguments is sometimes inevitable when using AFs for representation purposes. However, due to their very nature, the dialectical meaning of such technical arguments might be - ironically - debatable.

Not surprisingly, quite a number of generalisations of AFs have been proposed (for an overview we refer to Brewka, Polberg, \& Woltran, 2014). As one of the most general AF alternatives, the aforementioned abstract dialectical frameworks (ADFs) have emerged. In that formalism, not only arguments (called statements there) are abstract, but also links between arguments. While in AFs the links are necessarily attacks, in ADFs each statement has an associated acceptance condition - a Boolean function over its parent statements - that specifies exactly when the statement can be accepted. In this way, acceptance conditions ultimately express the meaning of links in an ADF. Even the restricted subclass of bipolar ADFs - where intuitively all links are supporting or attacking - is a proper generalisation of AFs, and a quite expressive one as we shall see in this paper.

ADFs could be called the lovechild of AFs and logic programs, since they combine intuitions and semantics from Dung-style abstract argumentation as well as logic programming (Brewka et al., 2013; Strass, 2013; Alviano \& Faber, 2015). While on the abstract level, ADFs are intended to function as "argumentation middleware" - a sufficiently expressive target formalism for translations from more concrete (application) formalisms. As part of the ADF success story, we just mention a reconstruction of the Carneades model of argument (Brewka \& Gordon, 2010), an instantiation of simple defeasible theories into ADFs (Strass, 2015a), and recent applications of ADFs for legal reasoning and reasoning with cases by Al-Abdulkarim, Atkinson, and Bench-Capon (2014, 2015).

In this paper, we approach abstract dialectical frameworks as knowledge representation formalisms, since they are used to represent knowledge about arguments and relationships between these arguments. We employ this view to analyse the representational capabilities of ADFs. Due to their roots in AFs and logic programs, we also compare the representational capabilities of those formalisms in the same setting. In this initial study we restrict ourselves to looking at two-valued semantics, more specifically the ADF (stable) model semantics, which corresponds to AF stable extension semantics, and the supported and stable model semantics for logic programs. We add propositional logic to have a well-known reference point. Analysing these precise formalisms additionally makes sense to us because the computational complexity of their respective model existence problems is the same (with one exception):

- for AFs, deciding stable extension existence is NP-complete (Dimopoulos, Nebel, \& Toni, 2002);
- for normal logic programs, deciding the existence of supported/stable models is NPcomplete (Bidoit \& Froidevaux, 1991; Marek \& Truszczyński, 1991);
- for ADFs, deciding the existence of (supported) models is NP-complete (Brewka et al., 2013), deciding the existence of stable models is $\Sigma_{2}^{P}$-complete for general ADFs (Brewka et al., 2013) and NP-complete for the subclass of bipolar ADFs (Strass \& Wallner, 2015);
- the propositional satisfiability problem is NP-complete.

In view of these almost identical complexities, we use an alternative measure of the expressiveness of a knowledge representation formalism $\mathcal{F}$ : "Given a set of two-valued interpretations, is there a knowledge base in $\mathcal{F}$ that has this exact model set?" This notion
lends itself straightforwardly to compare different formalisms (Gogic, Kautz, Papadimitriou, \& Selman, 1995):

Formalism $\mathcal{F}_{2}$ is at least as expressive as formalism $\mathcal{F}_{1}$ if and only if every knowledge base in $\mathcal{F}_{1}$ has an equivalent knowledge base in $\mathcal{F}_{2}$.

So here expressiveness is understood in terms of realisability, "What kinds of model sets can the formalism express?" (In model theory, this is known as definability.)

It is easy to see that propositional logic can express any set of two-valued interpretations, it is universally expressive. The same is easy (but less easy) to see for normal logic programs under supported model semantics. For normal logic programs under stable model semantics, it is clear that not all model sets can be expressed, since two different stable models are always incomparable with respect to the subset relation. ${ }^{1}$ In this paper, we study such expressiveness properties for all the mentioned formalisms under different semantics. It turns out that the languages form a more or less strict expressiveness hierarchy, with AFs at the bottom, ADFs and LPs under stable semantics higher up and ADFs and LPs under supported model semantics at the top together with propositional logic.

To show that a language $\mathcal{F}_{2}$ is at least as expressive as a language $\mathcal{F}_{1}$ we will mainly use two different techniques. In the best case, we can use a syntactic compact and faithful translation from knowledge bases of $\mathcal{F}_{1}$ to those of $\mathcal{F}_{2}$. Compact means that the translation does not change the vocabulary, that is, does not introduce new atoms. Faithful means that the translation exactly preserves the models of the knowledge base for respective semantics of the two languages. In the second best case, we assume the knowledge base of $\mathcal{F}_{1}$ to be given in the form of a set $X$ of desired models and construct a semantic realisation of $X$ in $\mathcal{F}_{2}$, that is, a knowledge base in $\mathcal{F}_{2}$ with model set precisely $X$. To show that language $\mathcal{F}_{2}$ is strictly more expressive than $\mathcal{F}_{1}$, we additionally have to present a knowledge base kb from $\mathcal{F}_{2}$ of which we prove that $\mathcal{F}_{1}$ cannot express the model set of kb.

Analysing the expressiveness of argumentation formalisms is a quite recent strand of work. Its ascent can be attributed to Dunne et al. (2014, 2015), who studied realisability for argumentation frameworks (allowing to introduce new arguments as long as they are never accepted). Likewise, Dyrkolbotn (2014) analysed AF realisability under projection (allowing to introduce new arguments) for three-valued semantics. Baumann, Dvořák, Linsbichler, Strass, and Woltran (2014) studied the expressiveness of the subclass of "compact" AFs, where each argument is accepted at least once. Finally, and most recently, Pührer (2015) analysed the realisability of three-valued semantics for ADFs. Previous more preliminary works include that of Brewka, Dunne, and Woltran (2011), who translated ADFs into AFs for the ADF model and AF stable extension semantics, however this translation introduces additional arguments and is therefore not compact; and ours (Strass, 2013), where we studied the syntactic intertranslatability of ADFs and LPs, but did not look at expressiveness or realisability.

The gain that is achieved by our analysis in this paper is not only that of increased clarity about fundamental properties of these knowledge representation languages - What can these formalisms express, actually? - but has several further applications. As Dunne et al. (2015) remarked, a major application is in constructing knowledge bases with the aim

[^0]of encoding a certain model set. As a necessary prerequisite to this, it must be known that the intended model set is realisable in the first place. For example, in a recent approach to revising argumentation frameworks (Coste-Marquis, Konieczny, Mailly, \& Marquis, 2014), the authors avoid this problem by assuming to produce a collection of AFs whose model sets in union produce the desired model set. While the work of Dunne et al. (2015) showed that this is indeed necessary in the case of AFs and stable extension semantics, our work shows that for ADFs under the model semantics, a single knowledge base (ADF) is always enough to realise any given model set. What is more, if we assume that the intended model set is given in the form of a propositional formula, then the size of the realising ADF is at most linear in the size of the formula. This is only one example - we will on several occasions also consider the sizes of realisations, as is not uncommon in logic-based AI (Darwiche \& Marquis, 2002; Lifschitz \& Razborov, 2006; French, van der Hoek, Iliev, \& Kooi, 2013; Shen \& Zhao, 2014). Indeed, representation size is a fundamental practical aspect of knowledge representation languages: universal expressiveness is of little use if the model sets to express require exponential-size knowledge bases even in the best case!

Of course, the fact that the languages we study have the same computational complexity means that there in principle exist polynomial intertranslations for the respective decision problems. But such intertranslations may involve the introduction of a polynomial number of new atoms. In theory, an increase from $n$ atoms to $n^{k}$ atoms for some $k>1$ is of no consequence. In practice, it has a profound impact: the number $n$ of atoms directly influences the search space that any implementation potentially has to cover. There, a step from $2^{n}$ to

$$
2^{n^{k}}=2^{n^{k-1} n}=\left(2^{n^{k-1}}\right)^{n}
$$

amounts to an exponential increase in search space size. Being able to realise a model set compactly, without new atoms, therefore attests that a formalism $\mathcal{F}$ has a certain basic kind of efficiency property, in the sense that the $\mathcal{F}$-realisation of a model set does not unnecessarily enlarge the search space of algorithms operating on it.

It might seem that it is a restricting assumption to view formalisms as sets $\mathcal{F}$ of knowledge bases kb where $\mathcal{F}$ is associated with a two-valued semantics. However, this language representation model is universal in the sense that it is just another way of expressing languages as sets of words over $\{0,1\}$. Using an $n$-element vocabulary $A_{n}=\left\{a_{1}, \ldots, a_{n}\right\}$, a binary word $w=x_{1} x_{2} \cdots x_{n}$ of length $n$ is encoded as the set $M_{w}=\left\{a_{i} \in A_{n} \mid x_{i}=1\right\} \subseteq A_{n}$. For example, using the vocabulary $A_{3}=\left\{a_{1}, a_{2}, a_{3}\right\}$, the binary word 101 of length 3 corresponds to the set $M_{101}=\left\{a_{1}, a_{3}\right\}$. Consequently, a set $L_{n}$ of words of length $n$ can be represented by a set $X_{L_{n}} \subseteq 2^{A_{n}}$ of subsets of $A_{n}: X_{L_{n}}=\left\{M_{w} \mid w \in L_{n}\right\}$. With the above example vocabulary, the word set $L_{3}=\{101,110,011\}$ is represented by the model set $X_{L_{3}}=\left\{\left\{a_{1}, a_{3}\right\},\left\{a_{1}, a_{2}\right\},\left\{a_{2}, a_{3}\right\}\right\}$. Conversely, each sequence $\left(X_{n}\right)_{n \geq 0}$ of sets with $X_{n} \subseteq 2^{A_{n}}$ uniquely determines a language $L=\bigcup_{n>0} L_{n}$ over $\{0,1\}$ : for each $n \in \mathbb{N}$, we have $L_{n}=\left\{w_{M} \mid M \in X_{n}\right\}$ with $w_{M}=x_{1} x_{2} \cdots x_{n}$ where for each $i \in\{1, \ldots, n\}, x_{i}=1$ if $a_{i} \in M$ and $x_{i}=0$ if $a_{i} \notin M$. In this paper we use "language" to refer to object-level languages while "formalism" refers to meta-level languages, such as propositional logic, argumentation frameworks, abstract dialectical frameworks, and logic programs.

Formally, the syntax of ADFs is defined via Boolean functions. However, we are interested in representations of ADFs. So we have to fix a representation of ADFs via fixing
a representation of Boolean functions. We choose to use (unrestricted) propositional formulas, as is customary in most of the literature (Brewka \& Woltran, 2010; Brewka et al., 2013; Polberg et al., 2013; Polberg, 2014; Gaggl \& Strass, 2014; Linsbichler, 2014; Strass \& Wallner, 2015; Pührer, 2015; Gaggl, Rudolph, \& Strass, 2015). Exceptions to this custom are the works of Brewka et al. (2011), who use Boolean circuits, and one of ours (Strass, 2013) where we used characteristic models (that is, used a representation that is equivalent to representing the formulas in disjunctive normal form). For the subclass of bipolar ADFs, yet no uniform representation exists, which is another question we address in this paper.

By propositional formulas over a vocabulary $A$ we mean formulas over the Boolean basis $\{\wedge, \vee, \neg\}$, that is, trees whose leaves (sinks) are atoms from $A$ or the logical constants true $T$ or false $\perp$, and internal nodes are either unary $(\neg)$ or binary $(\wedge, \vee)$. We also make occasional use of Boolean circuits, where "trees" above is replaced by "directed acyclic graphs"; in particular, we allow unbounded fan-in, that is, reusing sub-circuits. As usual, the depth of a formula (circuit) is the length of the longest path from the root to a leaf (sink). Figure 1 below shows formula and circuit examples of depth 3 .


Figure 1: Representing $(p \wedge \neg q) \vee(q \wedge \neg p)$ as a formula tree (left) and a circuit (right).

Analysing the expressive power and representation size of Boolean circuits is an established sub-field of computational complexity (Arora \& Barak, 2009). This has led to a number of language classes whose members can be recognised by Boolean circuits satisfying certain restrictions. We will need the class $\mathrm{AC}^{0}$, which contains all languages $L=\bigcup_{n \geq 0} L_{n}$ for which there exist $d, k \in \mathbb{N}$ such that for each $n \in \mathbb{N}$, there exists a Boolean circuit $\bar{C}_{n}$ of depth at most $d$ and size at most $n^{k}$ where the models of $C_{n}$ exactly express $L_{n} .{ }^{2}$ In other words, every language $L \in \mathrm{AC}^{0}$ can be recognised by a family of polynomial-size Boolean circuits of a fixed maximal depth that is independent of word length.

The paper proceeds as follows. We first define the notion of expressiveness (and succinctness) formally and then introduce the formalisms we will study. After reviewing several intertranslatability results for these languages, we step-wise obtain the results that lead to the expressiveness hierarchy, while at times also looking at representational efficiency. We finally show that allowing to linearly expand the vocabulary leads to a collapse of the hierarchy. The paper concludes with a discussion of possible future work.
2. To be more precise, for each $n \in \mathbb{N}$, the models of $C_{n}$ are exactly $X_{L_{n}}$, which in turn expresses $L_{n}$.

## 2. Background

We presume a finite set $A$ of atoms (statements, arguments), the vocabulary. A knowledge representation formalism interpreted over $A$ is then some set $\mathcal{F}$; a (two-valued) semantics for $\mathcal{F}$ is a mapping $\sigma: \mathcal{F} \rightarrow 2^{2^{A}}$ that assigns sets of two-valued models to knowledge bases $\mathrm{kb} \in \mathcal{F}$. (So $A$ is implicit in $\sigma$.) Strictly speaking, a two-valued interpretation is a mapping from the set of atoms into the two truth values true and false, but for technical ease we represent two-valued interpretations by the sets containing the atoms that are true. Below, we write $\sigma(\mathcal{F})=\{\sigma(\mathrm{kb}) \mid \mathrm{kb} \in \mathcal{F}\}$; intuitively, $\sigma(\mathcal{F})$ is the set of interpretation sets that formalism $\mathcal{F}$ can express, with any knowledge base whatsoever. For example, for $\mathcal{F}=\mathrm{PL}$ propositional logic and $\sigma=$ mod the usual model semantics, we have $\sigma(\mathrm{PL})=2^{2^{A}}$ since obviously any set of models is realisable in propositional logic. ${ }^{3}$ This leads us to compare different pairs of languages and semantics with respect to the semantics' range of models. Our concept of "formalism" concentrates on semantics and decidedly remains abstract. We first define the expressiveness relation among formalisms.

Definition 1. Let $A$ be a finite vocabulary, $\mathcal{F}_{1}, \mathcal{F}_{2}$ be formalisms that are interpreted over $A$ and $\sigma_{1}: \mathcal{F}_{1} \rightarrow 2^{2^{A}}$ and $\sigma_{2}: \mathcal{F}_{2} \rightarrow 2^{2^{A}}$ be two-valued semantics. We define

$$
\mathcal{F}_{1}^{\sigma_{1}} \leq_{e} \mathcal{F}_{2}^{\sigma_{2}} \quad \text { iff } \quad \sigma_{1}\left(\mathcal{F}_{1}\right) \subseteq \sigma_{2}\left(\mathcal{F}_{2}\right)
$$

Intuitively, formalism $\mathcal{F}_{2}$ under semantics $\sigma_{2}$ is at least as expressive as formalism $\mathcal{F}_{1}$ under semantics $\sigma_{1}$, because all model sets that $\mathcal{F}_{1}$ can express under $\sigma_{1}$ are also contained in those that $\mathcal{F}_{2}$ can produce under $\sigma_{2}$. (If the semantics are clear from the context we will omit them; this holds in particular for argumentation frameworks and propositional logic, where we only look at a single semantics.) As usual,

- $\mathcal{F}_{1}<_{e} \mathcal{F}_{2}$ iff $\mathcal{F}_{1} \leq_{e} \mathcal{F}_{2}$ and $\mathcal{F}_{2} Z_{e} \mathcal{F}_{1} ;$
- $\mathcal{F}_{1} \cong_{e} \mathcal{F}_{2}$ iff $\mathcal{F}_{1} \leq_{e} \mathcal{F}_{2}$ and $\mathcal{F}_{2} \leq_{e} \mathcal{F}_{1}$.

The relation $\leq_{e}$ is reflexive and transitive by definition, but not necessarily antisymmetric. That is, there might different formalisms $\mathcal{F}_{1} \neq \mathcal{F}_{2}$ that are equally expressive: $\mathcal{F}_{1} \cong \mathcal{F}_{2}$.

We next introduce the succinctness relation as defined by Gogic et al. (1995).
Definition 2. Let $A$ be a finite vocabulary; let $\mathcal{F}_{1}$ and $\mathcal{F}_{2}$ be formalisms that are interpreted over $A$, have size measures $\|\cdot\|_{1}$ and $\|\cdot\|_{2}$, and two-valued semantics $\sigma_{1}$ and $\sigma_{2}$, respectively. Define $\mathcal{F}_{1}^{\sigma_{1}} \leq_{s} \mathcal{F}_{2}^{\sigma_{2}}$ if and only if there is a $k \in \mathbb{N}$ such that for all $k b_{1} \in \mathcal{F}_{1}$ with $\sigma_{1}\left(\mathrm{~kb}_{1}\right) \in \sigma_{1}\left(\mathcal{F}_{1}\right) \cap \sigma_{2}\left(\mathcal{F}_{2}\right)$, there is a $\mathrm{kb}_{2} \in \mathcal{F}_{2}$ with $\sigma_{1}\left(\mathrm{~kb}_{1}\right)=\sigma_{2}\left(\mathrm{~kb}_{2}\right)$ and $\left\|\mathrm{kb}_{2}\right\|_{2} \leq\left\|\mathrm{kb}_{1}\right\|_{1}^{k}$.

Intuitively, $\mathcal{F}_{1}^{\sigma_{1}} \leq_{s} \mathcal{F}_{2}^{\sigma_{2}}$ means that $\mathcal{F}_{2}$ under $\sigma_{2}$ is at least as succinct as $\mathcal{F}_{1}$ under $\sigma_{1}$. Put another way, for $\mathcal{F}_{1}^{\sigma_{1}} \leq_{s} \mathcal{F}_{2}^{\sigma_{2}}$ to hold, any knowledge base from $\mathcal{F}_{1}$ with an equivalent counterpart in $\mathcal{F}_{2}$ must have an equivalent counterpart that is at most polynomially larger. Note that succinctness talks only about those model sets that both can express, so it is most meaningful when comparing languages that are equally expressive, that is, whenever
3. For a set $X \subseteq 2^{A}$ we can simply define $\varphi_{X}=\bigvee_{M \in X} \varphi_{M}$ with $\varphi_{M}=\bigwedge_{a \in M} a \wedge \bigwedge_{a \in A \backslash M} \neg a$ and clearly $\bmod \left(\varphi_{X}\right)=X$.
$\sigma_{1}\left(\mathcal{F}_{1}\right)=\sigma_{2}\left(\mathcal{F}_{2}\right)$. As usual, we define $\mathcal{F}_{1}<_{s} \mathcal{F}_{2}$ iff $\mathcal{F}_{1} \leq_{s} \mathcal{F}_{2}$ and $\mathcal{F}_{2} \not \mathbb{Z}_{s} \mathcal{F}_{1}$, and $\mathcal{F}_{1} \cong_{s} \mathcal{F}_{2}$ iff $\mathcal{F}_{1} \leq_{s} \mathcal{F}_{2}$ and $\mathcal{F}_{2} \leq_{s} \mathcal{F}_{1}$. The relation $\leq_{s}$ is reflexive, but not necessarily antisymmetric or transitive.

The final general definition is about formalisms polynomially expressing languages. Here, we already make use of the previously introduced bijection between interpretations and binary words and use the term "languages" to synonymously refer to both.

Definition 3. A formalism $\mathcal{F}$ can polynomially express a language $L=\bigcup_{n \geq 0} L_{n}$ under semantics $\sigma: \mathcal{F} \rightarrow 2^{2^{A}}$ if and only if there is a $k \in \mathbb{N}$ such that for each positive $n \in \mathbb{N}$ there is a knowledge base $\mathrm{kb}_{n} \in \mathcal{F}$ of that formalism such that $\sigma\left(\mathrm{kb}_{n}\right)=L_{n}$ and $\left\|\mathrm{kb}_{n}\right\| \in O\left(n^{k}\right)$.

We next introduce some specific object-level languages that we will use. First of all, the language Parity contains all odd-element subsets of the vocabulary. Formally, for $A_{n}=\left\{a_{1}, \ldots, a_{n}\right\}$ with $n \geq 1$ we have

$$
\text { PARITY }_{n}=\left\{M \subseteq A_{n}|\exists m \in \mathbb{N}:|M|=2 m+1\}\right.
$$

As explained before, then Parity $=\bigcup_{n \in \mathbb{N}, n \geq 1}$ Parity $_{n}$. It is a textbook result that Parity is expressible by polynomial-size propositional formulas (Jukna, 2012); for example, we can define $\Phi_{1}^{\text {Parity }}\left(a_{1}\right)=a_{1}$ and for $n \geq 2$ set

$$
\begin{aligned}
\Phi_{n}^{\text {Parity }}\left(a_{1}, \ldots, a_{n}\right)= & \left(\Phi_{n_{\downarrow}}^{\text {PARITY }}\left(a_{1}, \ldots, a_{n_{\downarrow}}\right) \wedge \neg \Phi_{n_{\uparrow}}^{\text {Parity }}\left(a_{n_{\downarrow}+1}, \ldots, a_{n}\right)\right) \vee \\
& \left(\neg \Phi_{n_{\downarrow}}^{\text {PARITY }}\left(a_{1}, \ldots, a_{n_{\downarrow}}\right) \wedge \Phi_{n_{\uparrow}}^{\text {PARITY }}\left(a_{n_{\downarrow}+1}, \ldots, a_{n}\right)\right)
\end{aligned}
$$

with $n_{\downarrow}=\left\lfloor\frac{n}{2}\right\rfloor$ and $n_{\uparrow}=\left\lceil\frac{n}{2}\right\rceil$. (This construction yields a formula of logarithmic depth and therefore polynomial size.) It is also a textbook result (although not nearly as easy to see) that Parity cannot be expressed by depth-bounded polynomial-size circuits, that is, Parity $\notin \mathrm{AC}^{0}$ (Jukna, 2012).

As another important class, threshold languages are defined for $n, k \in \mathbb{N}$ with $n \geq 1$ and $k \leq n$ :

$$
\text { Threshold }_{n, k}=\left\{M \subseteq A_{n}|k \leq|M|\}\right.
$$

That is, Threshold $_{n, k}$ contains all interpretations over $n$ atoms where at least $k$ atoms are true. The special case $k=\left\lceil\frac{n}{2}\right\rceil$ leads to the majority languages,

$$
\text { Majority }_{n}=\text { Threshold }_{n,\left\lceil\frac{n}{2}\right\rceil}
$$

that contain all interpretations where at least half of the atoms in the vocabulary are true.
We next introduce the particular knowledge representation languages we study in this paper. All will make use of a vocabulary $A$; the results of the paper are all considered parametric in such a given vocabulary.

### 2.1 Logic Programs

For a vocabulary $A$ we define not $A=\{$ not $a \mid a \in A\}$ and accordingly the set of literals over $A$ as $A^{ \pm}=A \cup$ not $A$. A normal logic program rule over $A$ is then of the form $a \leftarrow B$ where $a \in A$ and $B \subseteq A^{ \pm}$. The set $B$ is called the body of the rule, we abbreviate $B^{+}=B \cap A$ and
$B^{-}=\{a \in A \mid$ not $a \in B\}$. A logic program (LP) $P$ over $A$ is a set of logic program rules over $A$. An interpretation $M \subseteq A$ satisfies the body $B$ of a rule $a \leftarrow B \in P$ iff $B^{+} \subseteq M$ and $B^{-} \cap M=\emptyset . M$ is a supported model for $P$ iff $M=\{a \in A \mid a \leftarrow B \in P, M$ satisfies $B\}$. For a logic program $P$ we denote the set of its supported models by $s u(P)$. The intuition behind this semantics is that the atoms that are true in a model are all and only those that have some kind of support.

However, this support might be cyclic self-support. For instance, the logic program $\{a \leftarrow\{a\}\}$ has two supported models, $\emptyset$ and $\{a\}$, where the latter is undesired in many application domains. As an alternative, Gelfond and Lifschitz (1988) proposed the stable model semantics, that does not allow self-support: A set $M \subseteq A$ is a stable model for $P$ iff $M$ is the $\subseteq$-least supported model of $P^{M}$, where $P^{M}$ is obtained from $P$ by (1) eliminating each rule whose body contains a literal not $a$ with $a \in M$, and (2) deleting all literals of the form not a from the bodies of the remaining rules (Gelfond \& Lifschitz, 1988). We write $s t(P)$ for the set of stable models of $P$. It follows from the definition that $s t(P)$ is a $\subseteq$-antichain: for all $M_{1} \neq M_{2} \in \operatorname{st}(P)$ we have $M_{1} \nsubseteq M_{2}$. As size measure we define $\|a \leftarrow B\|=|B|+1$ for rules and $\|P\|=\sum_{r \in P}\|r\|$ for programs.

As an example, consider the vocabulary $A=\{a, b, c\}$ and over it the logic program $P=\{a \leftarrow\{b\}, b \leftarrow\{a\}, c \leftarrow\{$ not $a\}\}$. We find $s u(P)=\{\{c\},\{a, b\}\}$ and $s t(P)=\{\{c\}\}$.

### 2.2 Argumentation Frameworks

Dung (1995) introduced argumentation frameworks as pairs $F=(A, R)$ where $A$ is a set of (abstract) arguments and $R \subseteq A \times A$ a relation of attack between the arguments. The purpose of semantics for argumentation frameworks is to determine sets of arguments (called extensions) which are acceptable according to various standards. For a given extension $S \subseteq A$, the arguments in $S$ are considered to be accepted, those that are attacked by some argument in $S$ are considered to be rejected, and all others are neither, their status is undecided. We will only be interested in so-called stable extensions, sets $S$ of arguments that do not attack each other and attack all arguments not in the set. For stable extensions, each argument is either accepted or rejected by definition, thus the semantics is two-valued. More formally, a set $S \subseteq A$ of arguments is conflict-free iff there are no $a, b \in S$ with $(a, b) \in R$. A set $S$ is a stable extension for $(A, R)$ iff it is conflict-free and for all $a \in A \backslash S$ there is an argument $b \in S$ with $(b, a) \in R$. For an AF $F$, we denote the set of its stable extensions by $s t(F)$. Again, it follows from the definition of a stable extension that the set $s t(F)$ is always a $\subseteq$-antichain. The size of an argumentation framework $F=(A, R)$ is $\|F\|=|A|+|R|$.

For example, the AF $F=(\{a, b, c\},\{(a, b),(b, a),(b, c)\})$ can be visualised using the directed graph $a \rightarrow c \rightarrow c$ and has the set of stable extensions $s t(F)=\{\{a, c\},\{b\}\}$.

### 2.3 Abstract Dialectical Frameworks

An abstract dialectical framework is a tuple $D=(A, L, C)$ where $A$ is a set of statements (representing positions one can take or not take in a debate), $L \subseteq A \times A$ is a set of links (representing dependencies between the positions), $C=\left\{C_{a}\right\}_{a \in A}$ is a collection of total functions $C_{a}: 2^{\operatorname{par}(a)} \rightarrow\{\mathbf{t}, \mathbf{f}\}$, one for each statement $a \in A$. The function $C_{a}$ is called acceptance condition of $a$ and expresses whether $a$ can be accepted, given the acceptance
status of its parents $\operatorname{par}(a)$. In this paper, we represent each $C_{a}$ by a propositional formula $\varphi_{a}$ over $\operatorname{par}(a)$. As mentioned earlier, propositional formulas are built using negation $\neg$, conjunction $\wedge$ and disjunction $\vee$; connectives for material implication $\rightarrow$, logical equivalence $\leftrightarrow$ and exclusive disjunction $\leftrightarrow$ are regarded as abbreviations. To specify an acceptance condition, then, we take $C_{a}(M \cap \operatorname{par}(a))=\mathbf{t}$ to hold iff $M$ is a model for $\varphi_{a}, M \models \varphi_{a}$.

Brewka and Woltran (2010) introduced a useful subclass of ADFs: an ADF $D=(A, L, C)$ is bipolar iff all links in $L$ are supporting or attacking (or both). A link $(b, a) \in L$ is supporting in $D$ iff for all $M \subseteq \operatorname{par}(a)$, we have that $C_{a}(M)=\mathbf{t}$ implies $C_{a}(M \cup\{b\})=\mathbf{t}$. Symmetrically, a link $(b, a) \in L$ is attacking in $D$ iff for all $M \subseteq \operatorname{par}(a)$, we have that $C_{a}(M \cup\{b\})=\mathbf{t}$ implies $C_{a}(M)=\mathbf{t}$. If a link $(b, a)$ is both supporting and attacking then $b$ has no influence on $a$, the link is redundant (but does not violate bipolarity). We will sometimes use this circumstance when searching for ADFs; there we simply assume that $L=A \times A$, then links that are actually not needed can be expressed by acceptance conditions that make them redundant.

There are numerous semantics for ADFs; we will only be interested in two of them, (supported) models and stable models. A set $M \subseteq A$ is a model of $D$ iff for all $a \in A$ we find that $a \in M$ iff $C_{a}(M)=\mathbf{t}$. The definition of stable models is inspired by logic programming and slightly more complicated (Brewka et al., 2013). Define an operator by ${ }^{4}$

$$
\begin{aligned}
\Gamma_{D}(X, Y) & =(a c(X, Y), r e(X, Y)) \text { for } X, Y \subseteq A, \text { where } \\
a c(X, Y) & =\left\{a \in A \mid \forall Z \subseteq A: X \subseteq Z \subseteq A \backslash Y \Rightarrow C_{a}(Z)=\mathbf{t}\right\} \\
r e(X, Y) & =\left\{a \in A \mid \forall Z \subseteq A: X \subseteq Z \subseteq A \backslash Y \Rightarrow C_{a}(Z)=\mathbf{f}\right\}
\end{aligned}
$$

The intuition behind the operator is as follows: A pair ( $X, Y$ ) represents a partial interpretation of the set of statements where those in $X$ are accepted (true), those in $Y$ are rejected (false), and those in $A \backslash(X \cup Y)$ are neither. The operator checks for each statement $a$ whether all total interpretations that can possibly arise from $(X, Y)$ agree on their truth value for the acceptance condition of $a$. That is, if $a$ has to be accepted no matter how the statements in $A \backslash(X \cup Y)$ are interpreted, then $a \in \operatorname{acc}(X, Y)$. The set $\operatorname{rej}(X, Y)$ is defined symmetrically, so the pair $(\operatorname{acc}(X, Y), \operatorname{rej}(X, Y))$ constitutes a refinement of $(X, Y)$.

For $M \subseteq A$, the reduced ADF $D^{M}=\left(M, L^{M}, C^{M}\right)$ is defined by $L^{M}=L \cap M \times M$ and for each $a \in M$ setting $\varphi_{a}^{M}=\varphi_{a}[b / \perp: b \notin M]$, that is, replacing all $b \notin M$ by false in the acceptance formula of $a$. A model $M$ for $D$ is a stable model of $D$ iff the least fixpoint of the operator $\Gamma_{D^{M}}$ is given by $(M, \emptyset)$. As usual, $s u(D)$ and $s t(D)$ denote the respective model sets; while ADF models can be $\subseteq$-related, ADF stable models cannot. The size of an ADF $D$ over $A$ is given by $\|D\|=\sum_{a \in A}\left\|\varphi_{a}\right\|$; the size $\|\varphi\|$ of a formula $\varphi$ is the number of its nodes.

As an example ADF $D$, consider vocabulary $A=\{a, b, c\}$ and the acceptance formulas $\varphi_{a}=c, \varphi_{b}=c$, and $\varphi_{c}=a \leftrightarrow b$. While $D$ has a single supported model, $s u(D)=\{\{a, b, c\}\}$, we find $s t(D)=\emptyset$ since the atoms in the model support each other circularly.

### 2.4 Translations Between the Formalisms

We will review all known translations between the mentioned formalisms.

[^1]
### 2.4.1 From AFs to BADFs

Brewka and Woltran (2010) showed how to translate AFs into ADFs: For an AF $F=(A, R)$, define the ADF associated to $F$ as $D_{F}=(A, R, C)$ with $C=\left\{\varphi_{a}\right\}_{a \in A}$ and $\varphi_{a}=\bigwedge_{(b, a) \in R} \neg b$ for $a \in A$. Clearly, the resulting ADF is bipolar: parents are always attacking. Brewka and Woltran proved that this translation is faithful for the AF stable extension and ADF model semantics (Proposition 1). Brewka et al. (2013) later proved the same for the AF stable extension and ADF stable model semantics (Theorem 4). It is easy to see that the translation can be computed in polynomial time and induces at most a linear blowup.

### 2.4.2 From ADFs to PL

Brewka and Woltran (2010) also showed that ADFs under supported model semantics can be faithfully translated into propositional logic: when acceptance conditions of statements $a \in A$ are represented by propositional formulas $\varphi_{a}$, then the supported models of an ADF $D$ over $A$ are given by the classical propositional models of the formula set $\Phi_{D}=\left\{a \leftrightarrow \varphi_{a} \mid a \in A\right\}$.

### 2.4.3 From AFs to PL

In combination, the previous two translations yield a polynomial and faithful translation chain from AFs into propositional logic: $\Phi_{(A, R)}=\left\{a \leftrightarrow\left(\bigwedge_{(b, a) \in R} \neg b\right) \mid a \in A\right\}$.

### 2.4.4 From ADFs to LPs

In earlier work (Strass, 2013), we showed that ADFs can be faithfully translated into normal logic programs. For an ADF $D=(A, L, C)$, its standard LP is

$$
P_{D}=\left\{a \leftarrow(M \cup \operatorname{not}(\operatorname{par}(a) \backslash M)) \mid a \in A, C_{a}(M)=\mathbf{t}\right\}
$$

It follows from Lemma 3.14 of Strass (2013) that this translation preserves the supported model semantics. The translation is size-preserving for the acceptance condition representation of Strass (2013) via characteristic models; when representing acceptance conditions via propositional formulas, this cannot be guaranteed as we will show later. ${ }^{5}$

### 2.4.5 From AFs to LPs

The translation chain from AFs to ADFs to LPs is compact, and faithful for AF stable semantics and LP stable semantics (Osorio, Zepeda, Nieves, \& Cortés, 2005), and AF stable semantics and LP supported semantics (Strass, 2013). It is size-preserving since the single rule for each atom contains all attackers once: $P_{(A, R)}=\{a \leftarrow\{n o t b \mid(b, a) \in R\} \mid a \in A\}$.

[^2]
### 2.4.6 From LPs тo PL

It is well-known that logic programs under supported model semantics can be translated to propositional logic (Clark, 1978). A logic program $P$ becomes the propositional theory $\Phi_{P}$,

$$
\Phi_{P}=\left\{a \leftrightarrow \varphi_{a} \mid a \in A\right\} \quad \text { where } \quad \varphi_{a}=\bigvee_{a \leftarrow B \in P}\left(\bigwedge_{b \in B^{+}} b \wedge \bigwedge_{b \in B^{-}} \neg b\right) \quad \text { for } a \in A .
$$

For the stable model semantics, additional formulas have to be added, but the extended translation works all the same (Lin \& Zhao, 2004).

### 2.4.7 From LPs to ADFs

The Clark completion of a normal logic program directly yields an equivalent ADF over the same signature (Brewka \& Woltran, 2010). Clearly the translation is computable in polynomial time and the blowup (with respect to the original logic program) is at most linear. The resulting translation is faithful for the supported model semantics, which follows from Lemma 3.16 of Strass (2013).

### 2.5 Representing Bipolar Boolean Functions

While bipolarity has hitherto predominantly been defined and used in the context of ADFs (Brewka \& Woltran, 2010), it is easy to define the concept for Boolean functions in general. Let $A$ be a set of atoms and $f: 2^{A} \rightarrow\{\mathbf{t}, \mathbf{f}\}$ be a Boolean function. An atom $a \in A$ is supporting iff for all $M \subseteq A, f(M)=\mathbf{t}$ implies $f(M \cup\{a\})=\mathbf{t}$; we then write $a \in \sup (f)$. An atom $a \in A$ is attacking iff for all $M \subseteq A, f(M)=\mathbf{f}$ implies $f(M \cup\{a\})=\mathbf{f}$; we then write $a \in \operatorname{att}(f)$. A Boolean function $f: 2^{A} \rightarrow\{\mathbf{t}, \mathbf{f}\}$ is semantically bipolar iff each $a \in A$ is supporting or attacking or both. Throughout the paper, we will sometimes take a Boolean function to be given by an interpretation set and then say that the set is bipolar.

We will now define bipolar propositional formulas for representing bipolar ADFs. This is important not only for our study, but also since (for three-valued semantics), bipolarity is the key to BADFs' low complexity in comparison to general ADFs (Strass \& Wallner, 2015). Up to now, we usually assumed that to specify a bipolar ADF, in addition to statements, links and acceptance conditions, the user specifies for each link whether it is supporting or attacking (Strass \& Wallner, 2015). Here we introduce an arguably simpler way, where support and attack is represented in the syntax of the propositional formula encoding the acceptance function.

Formally, the polarity of an atom $a \in A$ in a formula is determined by the number of negations on the path from the root of the formula tree to the atom. The polarity is positive if the number is even and negative if the number is odd.

Definition 4. A propositional formula $\varphi$ over $A$ is syntactically bipolar if and only if no atom $a \in A$ occurs both positively and negatively in $\varphi$.
Recall that we only use formulas over the basis $\{\wedge, \vee, \neg\}$ and thus there are no "hidden" negations, e.g. from material implication. For formulas in negation normal form (that is, where negation is only applied to atomic formulas), the polarities of the atoms can be read off the formula directly.

We will now address the question how to represent bipolar Boolean functions. Clearly all Boolean functions can be represented by propositional formulas; we modify this construction later and thus reproduce it here: for a Boolean function $f: 2^{A} \rightarrow\{\mathbf{t}, \mathbf{f}\}$, its associated formula is

$$
\begin{equation*}
\varphi_{f}=\bigvee_{M \subseteq A, f(M)=\mathbf{t}} \varphi_{M} \quad \text { with } \quad \varphi_{M}=\bigwedge_{a \in M} a \wedge \bigwedge_{a \in A \backslash M} \neg a \tag{1}
\end{equation*}
$$

That is, each $\varphi_{M}$ has exactly one model $M$, and $\varphi_{f}$ enumerates those models.
So in particular, all bipolar Boolean functions can be represented by propositional formulas as well. However, this only guarantees us the existence of such representations but gives us no way to actually obtain them. Our first fundamental result shows how we can construct a syntactically bipolar propositional formula from a given semantically bipolar Boolean function. The converse is straightforward, and thus the two notions of bipolarity are closely related. For a formula $\varphi$, its associated Boolean function $f_{\varphi}$ returns $\mathbf{t}$ if and only if it gets as input a model of $\varphi$.

Theorem 1. Let $A$ be a set of atoms.

1. For each syntactically bipolar formula $\varphi$ over $A$, its Boolean function $f_{\varphi}$ is semantically bipolar.
2. For each semantically bipolar Boolean function $f: 2^{A} \rightarrow\{\mathbf{t}, \mathbf{f}\}$, a syntactically bipolar formula $\psi_{f}$ with $f_{\psi_{f}}=f$ is given by

$$
\begin{equation*}
\psi_{f}=\bigvee_{\substack{M \subseteq A, f(M)=\mathbf{t}}} \psi_{M} \quad \text { with } \quad \psi_{M}=\bigwedge_{\substack{a \in M, a \notin \operatorname{att}(f)}} a \wedge \bigwedge_{\substack{a \in A \backslash M, a \notin \sup (f)}} \neg a \tag{2}
\end{equation*}
$$

Proof. 1. Obvious: every atom occurring only positively is supporting, every atom occurring only negatively is attacking.
2. Let $f: 2^{A} \rightarrow\{\mathbf{t}, \mathbf{f}\}$ be semantically bipolar. Note first that by (2), for any $M \subseteq A$ we have $\models \varphi_{M} \rightarrow \psi_{M}$. It is easy to see that $\psi_{f}$ is syntactically bipolar: Since $f$ is semantically bipolar, each $a \in A$ is: (1) attacking and not supporting, then it occurs only negatively in $\psi_{f}$; or (2) supporting and not attacking, then it occurs only positively in $\psi_{f}$; or (3) supporting and attacking, then it does not occur in $\psi_{f}$. It remains to show that $f_{\psi_{f}}=f$; we show $\models \varphi_{f} \leftrightarrow \psi_{f}$.
$\vDash \varphi_{f} \rightarrow \psi_{f}$ : Let $v: A \rightarrow\{\mathbf{t}, \mathbf{f}\}$ with $v\left(\varphi_{f}\right)=\mathbf{t}$. Then there is an $M \subseteq A$ such that $f(M)=\mathbf{t}$ and $v\left(\varphi_{M}\right)=\mathbf{t}$. (Clearly $v=v_{M}$.) By $\models \varphi_{M} \rightarrow \psi_{M}$ we get $v\left(\psi_{M}\right)=\mathbf{t}$ and thus $v\left(\psi_{f}\right)=\mathbf{t}$.
$\models \psi_{f} \rightarrow \varphi_{f}$ : For each model $v$ of $\psi_{f}$, there is an $M \subseteq A$ with $f(M)=\mathbf{t}$ such that $v\left(\psi_{M}\right)=\mathbf{t}$. To show that each model of $\psi_{f}$ is a model of $\varphi_{f}$, we show that for all $M \subseteq A$ with $f(M)=\mathbf{t}$, each model $v$ of $\psi_{M}$ is a model of $\varphi_{f}$. Let $|A|=n$. Then each $\varphi_{M}$ contains exactly $n$ literals. For the corresponding $\psi_{M}$ there is a $k \in \mathbb{N}$ with $0 \leq k \leq n$ such that $\psi_{M}$ contains exactly $n-k$ literals. For two
interpretations $v_{1}: A \rightarrow\{\mathbf{t}, \mathbf{f}\}$ and $v_{2}: A \rightarrow\{\mathbf{t}, \mathbf{f}\}$, define the difference between them as $\delta\left(v_{1}, v_{2}\right)=\left\{a \in A \mid v_{1}(a) \neq v_{2}(a)\right\}$. (Note that for $|A|=n$ we always have $\left|\delta\left(v_{1}, v_{2}\right)\right| \leq n$.) We will use induction on $k$ to show the following: for each $M \subseteq A$ with $f(M)=\mathbf{t}$, each $v: A \rightarrow\{\mathbf{t}, \mathbf{f}\}$ with $v\left(\psi_{M}\right)=\mathbf{t}$ and $\left|\delta\left(v, v_{M}\right)\right|=k$ we find that $v\left(\varphi_{f}\right)=\mathbf{t}$. This covers all models $v$ of $\psi_{f}\left(\right.$ since $\left.\left|\delta\left(v, v_{M}\right)\right| \leq|A|\right)$ and thus establishes the claim.
$k=0: \delta\left(v, v_{M}\right)=\emptyset$ implies $v=v_{M}$ whence $v\left(\varphi_{f}\right)=v_{M}\left(\varphi_{f}\right)=v_{M}\left(\varphi_{M}\right)=\mathbf{t}$ by definition of $\varphi_{M}$ and $\varphi_{f}$.
$k \rightsquigarrow k+1:$ Let $M \subseteq A$ with $f(M)=\mathbf{t}$, and $v: A \rightarrow\{\mathbf{t}, \mathbf{f}\}$ with $v\left(\psi_{M}\right)=\mathbf{t}$ and $\left|\delta\left(v, v_{M}\right)\right|=k+1$. Since $k+1>0$, there is some $a \in \delta\left(v, v_{M}\right)$, that is, an $a \in A$ with $v(a) \neq v_{M}(a)$.
(a) $a$ is supporting and not attacking. Then necessarily $v(a)=\mathbf{t}$. (If $v(a)=\mathbf{f}$, then $v_{M}(a) \neq v(a)$ implies $v_{M}(a)=\mathbf{t}$, that is, $a \in M$ whence $\left\{\psi_{M}\right\} \models a$ and $v\left(\psi_{M}\right)=\mathbf{f}$, contradiction.) Define the interpretation $w: A \rightarrow\{\mathbf{t}, \mathbf{f}\}$ such that $w(a)=\mathbf{f}$ and $w(c)=v(c)$ for $c \in A \backslash\{a\}$. Clearly $\delta(v, w)=\{a\}$ and $\left|\delta\left(w, v_{M}\right)\right|=k$. Hence the induction hypothesis applies to $w$ and $w\left(\varphi_{f}\right)=\mathbf{t}$. Now $w(a)=\mathbf{f}, v(a)=\mathbf{t}$ and $w\left(\varphi_{f}\right)=\mathbf{t}$. Since $a$ is supporting, also $v\left(\varphi_{f}\right)=\mathbf{t}$.
(b) $a$ is attacking and not supporting. Symmetric to the opposite case above.
(c) $a$ is both supporting and attacking. Define interpretation $w: A \rightarrow\{\mathbf{t}, \mathbf{f}\}$ such that $w(a)=v_{M}(a)$ and $w(c)=v(c)$ for $c \in A \backslash\{a\}$. It follows that $\left|\delta\left(w, v_{M}\right)\right|=k$, whence the induction hypothesis applies to $w$ and $w\left(\varphi_{f}\right)=\mathbf{t}$. Since $a$ is both supporting and attacking (thus redundant), we get that $v\left(\varphi_{f}\right)=w\left(\varphi_{f}\right)=\mathbf{t}$.

This result paves the way for analysing the succinctness of bipolar ADFs, since now we have a quite natural way of representing them.

## 3. Relative Expressiveness

We now analyse and compare the relative expressiveness of argumentation frameworks (AFs), (bipolar) abstract dialectical frameworks ((B)ADFs), normal logic programs (LPs) and propositional logic (PL). We first look at the different families of semantics - supported and stable models - in isolation and afterwards combine the results for the two semantics. For the formalisms $\mathcal{F} \in\{$ ADF, LP $\}$ that have both supported and stable semantics, we will indicate the semantics $\sigma$ via a superscript as in Definition 1 . For AFs we only consider the stable semantics, as this is (to date) the only semantics for AFs where all interpretations are guaranteed to map all arguments to either true (accepted) or false (rejected, i.e. attacked by an accepted argument). For propositional logic PL we consider the usual model semantics.

With the syntactic translations we reviewed in the previous section, we currently have the following expressiveness relationships. For the supported semantics,

$$
\mathrm{AF} \leq_{e} \mathrm{BADF}^{s u} \leq_{e} \mathrm{ADF}^{s u} \cong_{e} \mathrm{LP}^{s u} \leq_{e} \mathrm{PL}
$$

and for the stable semantics,

$$
\mathrm{AF} \leq_{e} \mathrm{LP}^{s t}<_{e} \mathrm{PL} \text { and } \mathrm{AF} \leq_{e} \mathrm{BADF}^{s t} \leq_{e} \mathrm{ADF}^{s t}<_{e} \mathrm{PL}
$$

Note that $\mathrm{LP}^{s t}<_{e} \mathrm{PL}$ and $\mathrm{ADF}^{s t}<_{e} \mathrm{PL}$ hold since sets of stable models have an antichain property, in contrast to model sets of propositional logic.

For the succinctness relation, we have

$$
\mathrm{AF} \leq_{s} \mathrm{BADF}^{s u} \leq_{s} \mathrm{ADF}^{s u} \leq_{s} \mathrm{PL} \text { and } \mathrm{LP}^{s u} \leq_{s} \mathrm{ADF}^{s u}
$$

### 3.1 Supported Semantics

As depicted above, we know that expressiveness from AFs to propositional logic does not decrease. However, it is not yet clear if any of the relationships is strict. In what follows we will show that two of them are strict, working our way top-down from most to least expressive.

### 3.1.1 ADF vs. PL

We first show that ADFs can realise any set of models by showing how a given propositional formula can be used to construct an equivalent ADF of linear size. ${ }^{6}$

Theorem 2. $P L \leq_{e} A D F^{s u}$ and $P L \leq_{s} A D F^{s u}$.
Proof. Let $\psi$ be a propositional formula over vocabulary $A$. Define the ADF $D_{\psi}$ over $A$ by setting, for all $a \in A$,

$$
\varphi_{a}=a \leftrightarrow \psi=(a \wedge \psi) \vee(\neg a \wedge \neg \psi)
$$

Thus $\left\|\varphi_{a}\right\| \in O(\|\psi\|)$, whence $\left\|D_{\psi}\right\| \in O(|A| \cdot\|\psi\|)$. It remains to show $\operatorname{su}\left(D_{\psi}\right)=\bmod (\psi)$. Recall that for any ADF $D$ over $A, s u(D)=\bmod \left(\Phi_{D}\right)$ for $\Phi_{D}=\bigwedge_{a \in A}\left(a \leftrightarrow \varphi_{a}\right)$. Applying the definition of $\varphi_{a}$ in $D_{\psi}$ yields

$$
\Phi_{D_{\psi}}=\bigwedge_{a \in A}(a \leftrightarrow(a \leftrightarrow \psi))
$$

Now for any $a \in A$, the formula $(a \leftrightarrow(a \leftrightarrow \psi))$ is equivalent to $\psi$. (The proof is by case distinction on $a$.) Thus $\Phi_{D_{\psi}}$ is equivalent to $\bigwedge_{a \in A} \psi$, that is, to $\psi$, and it follows that $s u\left(D_{\psi}\right)=\bmod \left(\Phi_{D_{\psi}}\right)=\bmod (\psi)$.

For example, consider the vocabulary $A=\{a, b\}$ and the propositional formula $\psi=a \wedge b$. The canonical construction above yields ADF $D_{\psi}$ with acceptance formulas $\varphi_{a}=a \leftrightarrow(a \wedge b)$ and $\varphi_{b}=b \leftrightarrow(a \wedge b)$. Now we have:

$$
\varphi_{a}=a \leftrightarrow(a \wedge b)=(a \rightarrow(a \wedge b)) \wedge((a \wedge b) \rightarrow a) \equiv \neg a \vee(a \wedge b) \equiv \neg a \vee b
$$

Intuitively, $\varphi_{a}=\neg a \vee b$ expresses that $a$ cannot be false, and is true if $b$ is true. By a symmetrical argument, the acceptance formula of $b$ is equivalent to $\neg b \vee a$. It is readily checked that $s u\left(D_{\psi}\right)=\{\{a, b\}\}$ as desired. Since we know from Section 2.4.2 that the converse translation is also possible ( $\mathrm{ADF}^{s u} \leq_{s} \mathrm{PL}$ ), we get the following.

Corollary 3. $P L \cong_{s} A D F^{s u}$
6. If we consider the vocabulary $A$ to be part of the input, the size increase is quadratic.

When the acceptance conditions are written as propositional formulas, the construction to realise $X \subseteq 2^{A}$ in the proof of Theorem 2 defines a space-efficient equivalent of

$$
\varphi_{a}=\bigvee_{M \in X, a \in M} \varphi_{M} \vee \bigvee_{M \subseteq A, M \notin X, a \notin M} \varphi_{M}
$$

as acceptance formula of $a$, where $\varphi_{M}$ is as in Footnote 3.

### 3.1.2 ADF vs. LP

Since ADFs under supported semantics can be faithfully translated into logic programs, which can be likewise further translated to propositional logic, we have the following.
Corollary 4. $A D F^{s u} \cong_{e} L P^{s u} \cong_{e} P L$
However, this does not extend to the succinctness relation, as logic programs stipulate a particular syntactic form that is essentially a fixed-depth circuit. More specifically, it is easy to see that any language that is polynomially expressible by normal logic programs under supported semantics is in $A C^{0}$. For the stable semantics of so-called canonical logic programs, this has recently been shown by Shen and Zhao (2014) (Proposition 2.1). The case we are interested in (supported semantics) works similarly, but we still present the proof for completeness. The main technical result towards proving that is a lemma showing how to turn a logic program into an equivalent Boolean circuit of a fixed depth.
Lemma 5. For every normal logic program $P$, there exists a circuit $C_{P}$ over the basis $\{\neg, \wedge, \vee\}$ such that:

1. $C_{P}$ accepts all and only the supported models of $P$,
2. the size of $C_{P}$ is linear the size of $P$,
3. $C_{P}$ has depth 4.

Proof. Let $A=\left\{a_{1}, \ldots, a_{n}\right\}$ be the vocabulary of $P$, and its Clark completion be $\Phi_{P}=$ $\left\{a_{i} \leftrightarrow \psi_{i} \mid a_{i} \in A\right\}$ where the $\psi_{i}$ are DNFs over literals from $A$. Clearly the circuit for $\Phi_{P}$ must compute $C_{P}=\bigwedge_{a_{i} \in A}\left(a_{i} \leftrightarrow \psi_{i}\right)$ where $a_{i} \leftrightarrow \psi_{i}$ can be replaced by $\left(\neg a_{i} \vee \psi_{i}\right) \wedge\left(a_{i} \vee \neg \psi_{i}\right)$ with $\neg \psi_{i}$ a CNF over literals from $A$. The construction can be depicted as follows, where the inner layers are shown for one $i$ only, and dotted lines represent potential edges.


Now (1) follows since $s u(P)=\bmod \left(\Phi_{P}\right)$ and $C_{P}$ accepts all and only the models of $\Phi_{P}$. For (2), if $P$ contains $m=|P|$ rules, then $m \leq\|P\|$ and the total number of inner gates is bounded by $n(2 m+3) \leq n(2 \cdot\|P\|+3)$. (3) is clear.

While the statement of Lemma 5 is actually much stronger and gives a constant upper bound of the resulting circuit depth for arbitrarily-sized logic programs, it readily follows that the set of polynomially logic-program expressible languages is a subset of the languages expressible by alternating Boolean circuits with unbounded fan-in and constant depth.

Proposition 6. If $L$ is polynomially expressible by normal logic programs under supported semantics, then $L \in \mathrm{AC}^{0}$.

It follows immediately that normal logic programs cannot polynomially express the language Parity. ${ }^{7}$ This is the supported-semantics counterpart of Theorem 3.1 in (Shen \& Zhao, 2014).

Corollary 7. Parity has no polynomial size normal logic program representation.
Proof. By Proposition 6 and Parity $\notin \mathrm{AC}^{0}$ (Jukna, 2012).
It follows that propositional logic is strictly more succinct than normal logic programs under supported semantics.

Corollary 8. $P L \not \mathbb{Z}_{s} L P^{s u}$ and thus $L P^{s u}<_{s} P L$.
From our considerations since Theorem 2, it follows that if $\psi$ has a "small" conjunctive normal form (a conjunction of clauses) and disjunctive normal form (disjunction of monomials) representation, then there is also a "small" normal logic program representation for $\bmod (\psi)$.

### 3.1.3 ADF vs. BADF

It is quite obvious that the canonical ADF constructed in Theorem 2 is not bipolar, since $a$ as well as every atom mentioned by $\psi$ occurs both positively and negatively in $\varphi_{a}$. This raises the question whether the construction can be adapted to bipolar ADFs.

It turns out that the subclass of bipolar ADFs is strictly less expressive. Towards the proof of this result we start out with a new concept: that of the conjugate of a model set with respect to an atom. This concept will be used to characterise ADF realisability and precisely captures the "if-and-only-if part" of ADFs' supported model semantics: From the translation of an ADF $D$ into propositional logic (cf. Section 2.4.2) we can see that the result is basically a conjunction of equivalences: $\phi_{D}=\bigwedge_{a \in A}\left(a \leftrightarrow \varphi_{a}\right)$. While the conjunction part will be captured by set intersection, the conjugate will capture the equivalence part.

Definition 5. Let $A$ be a vocabulary, $X \subseteq 2^{A}$ and $a \in A$. The $a$-conjugate of $X$ is the set

$$
\langle a\rangle(X)=\{M \subseteq A \mid M \in X, a \in M\} \cup\{M \subseteq A \mid M \notin X, a \notin M\}
$$

7. Logic programs under supported models are universally expressive, so they can express Parity, just not in polynomial size.

Alternatively, we could write $\langle a\rangle(X)=\{M \subseteq A \mid M \in X \leftrightarrow a \in M\}$. Intuitively, $\langle a\rangle(X)$ contains all interpretations $M$ where containment of $a$ in $M$ coincides exactly with containment of $M$ in $X$. Formulated in terms of propositional formulas, if $X$ is the model set of formula $\varphi$ over $A$, then $\langle a\rangle(X)$ is the model set of formula $a \leftrightarrow \varphi$. Note that the vocabulary $A$ is implicit in the conjugate function.

Example 1. Consider the vocabulary $A_{2}=\{a, b\}$. The functions $\langle a\rangle(\cdot)$ and $\langle b\rangle(\cdot)$ operate on the set $2^{2^{A_{2}}}$ of interpretation sets over $A_{2}$ and are shown in Table 1.

| $\varphi$ | $\langle a\rangle(\varphi)$ | $\langle b\rangle(\varphi)$ |
| :---: | :---: | :---: |
| $\perp$ | $\neg a$ | $\neg b$ |
| $\neg a \wedge \neg b$ | $\neg a \wedge b$ | $a \wedge \neg b$ |
| $a \wedge \neg b$ | $\neg a \vee \neg b$ | $\neg a \wedge \neg b$ |
| $\neg a \wedge b$ | $\neg a \wedge \neg b$ | $\neg a \vee \neg b$ |
| $a \wedge b$ | $a \rightarrow b$ | $b \rightarrow a$ |
| $a$ | $\top$ | $a \leftrightarrow b$ |
| $b$ | $a \leftrightarrow b$ | $\top$ |
| $\neg a$ | $\perp$ | $a \leftrightarrow b$ |
| $\neg b$ | $a \leftrightarrow b$ | $\perp$ |
| $a \leftrightarrow b$ | $\neg b$ | $\neg a$ |
| $a \leftrightarrow b$ | $b$ | $a$ |
| $a \vee b$ | $b \rightarrow a$ | $a \rightarrow b$ |
| $\neg a \vee \neg b$ | $a \wedge \neg b$ | $\neg a \wedge b$ |
| $a \rightarrow b$ | $a \wedge b$ | $a \vee b$ |
| $b \rightarrow a$ | $a \vee b$ | $a \wedge b$ |
| $\top$ | $a$ | $b$ |

Table 1: Conjugation functions for $A_{2}=\{a, b\}$. Interpretation sets are represented using formulas over $A_{2}$, and connective " $\leftrightarrow$ " denotes exclusive disjunction XOR.

For two-valued ADF semantics, this conjugation function plays an essential semantical role, since it provides the "bridge" between models of the acceptance functions and models of the ADF. But it is also interesting in itself: We first show some properties of the conjugation function associated to an atom, since some of them will be used in the proof later on. First of all, it is an involution, that is, its own inverse (and thus in particular a bijection). Next, it is compatible with the complement operation (logical negation on the formula level). Finally, it also preserves the evenness of the cardinality of the input set.

Proposition 9. Let $A$ be a vocabulary, $X \subseteq 2^{A}$ and $a \in A$.

1. $\langle a\rangle(\langle a\rangle(X))=X$.
(involution)
2. $2^{A} \backslash\langle a\rangle(X)=\langle a\rangle\left(2^{A} \backslash X\right)$.
(compatible with negation)
3. $|X|$ is even iff $|\langle a\rangle(X)|$ is even.
(preserves evenness)
Proof. Let $|A|=n, X \subseteq 2^{A}$ and $a \in A$.
4. Let $M \subseteq A$. We have

$$
\begin{aligned}
M \in\langle a\rangle(\langle a\rangle(X)) & \text { iff } M \in\langle a\rangle(X) \leftrightarrow a \in M \\
& \text { iff }(M \in X \leftrightarrow a \in M) \leftrightarrow a \in M \\
& \text { iff } M \in X \leftrightarrow(a \in M \leftrightarrow a \in M) \\
& \text { iff } M \in X
\end{aligned}
$$

2. Denote

$$
\begin{aligned}
& S_{\in, \epsilon}=\{M \subseteq A \mid M \in X, a \in M\} \\
& S_{\in, \notin}=\{M \subseteq A \mid M \in X, a \notin M\} \\
& S_{\notin, \epsilon}=\{M \subseteq A \mid M \notin X, a \in M\} \\
& S_{\notin, \notin}=\{M \subseteq A \mid M \notin X, a \notin M\}
\end{aligned}
$$

and observe that

$$
\begin{aligned}
2^{A} & =S_{\epsilon, \epsilon} \uplus S_{\epsilon, \notin} \uplus S_{\notin, \epsilon} \uplus S_{\notin, \notin} \\
X & =S_{\in, \epsilon} \uplus S_{\in, \notin} \\
\langle a\rangle(X) & =S_{\in, \epsilon} \uplus S_{\notin, \notin}
\end{aligned}
$$

where $\uplus$ denotes disjoint union. Now

$$
\begin{aligned}
2^{A} \backslash\langle a\rangle(X) & =2^{A} \backslash\left(S_{\in, \epsilon} \uplus S_{\notin, \notin}\right) \\
& =S_{\in, \notin \uplus S_{\notin, \epsilon}} \\
& =\{M \subseteq A \mid M \in X, a \notin M\} \uplus\{M \subseteq A \mid M \notin X, a \in M\} \\
& =\left\{M \subseteq A \mid M \notin 2^{A} \backslash X, a \notin M\right\} \uplus\left\{M \subseteq A \mid M \in 2^{A} \backslash X, a \in M\right\} \\
& =\langle a\rangle\left(2^{A} \backslash X\right)
\end{aligned}
$$

3. We show that $|X|+|\langle a\rangle(X)|$ is even. Firstly,

$$
S_{\in, \notin} \uplus S_{\notin, \notin}=\{M \subseteq A \mid a \notin M\}=2^{A \backslash\{a\}}
$$

whence $\left|S_{\epsilon, \notin}\right|+\left|S_{\notin, \notin}\right|=2^{n-1}$. Thus

$$
\begin{aligned}
|X|+|\langle a\rangle(X)| & =\left(\left|S_{\epsilon, \epsilon}\right|+\left|S_{\epsilon, \notin}\right|\right)+\left(\left|S_{\epsilon, \epsilon}\right|+\left|S_{\notin, \notin}\right|\right) \\
& =2 \cdot\left|S_{\in, \epsilon}\right|+\left|S_{\epsilon, \notin}\right|+\left|S_{\notin, \notin}\right| \\
& =2 \cdot\left|S_{\in, \epsilon}\right|+2^{n-1}
\end{aligned}
$$

is even.

For our current purpose of characterising the expressiveness of bipolar ADFs, we now use the concept of conjugation to make ADF realisability for the model semantics slightly more accessible. We show that each ADF realisation of a model set $X$ over an $n$-element vocabulary $A$ can equivalently be characterised by an $n$-tuple $\left(Y_{1}, \ldots, Y_{n}\right)$ of supersets of $X$ whose intersection is exactly $X$. The crux of the proof of this result is how the acceptance conditions of the realising ADF and the $Y_{i}$ are related through the conjugation function.

Proposition 10. Let $A=\left\{a_{1}, \ldots, a_{n}\right\}$ be a vocabulary and $X \subseteq 2^{A}$ be a set of interpretations. Denote an ADF over $A$ by the sequence $\left(\varphi_{1}, \ldots, \varphi_{n}\right)$ of its acceptance formulas (for each $i \in\{1, \ldots, n\}$, formula $\varphi_{i}$ over $A$ is the acceptance formula of $a_{i}$ ), and further define

$$
\begin{aligned}
C_{X} & =\left\{\left(\bmod \left(\varphi_{1}\right), \ldots, \bmod \left(\varphi_{n}\right)\right) \mid \operatorname{su}\left(\varphi_{1}, \ldots, \varphi_{n}\right)=X\right\} \\
Y_{X} & =\left\{\left(Y_{1}, \ldots, Y_{n}\right) \mid Y_{1}, \ldots, Y_{n} \subseteq 2^{A},\left(\bigcap_{i=1}^{n} Y_{i}\right)=X\right\}
\end{aligned}
$$

The sets $C_{X}$ and $Y_{X}$ are in one-to-one correspondence; in particular $\left|C_{X}\right|=\left|Y_{X}\right|$.
Proof. We provide a bijection between $C_{X}$ and $Y_{X}$. Consider the function

$$
f:\left(2^{2^{A}}\right)^{n} \rightarrow\left(2^{2^{A}}\right)^{n} \quad \text { with } \quad\left(B_{1}, \ldots, B_{n}\right) \mapsto\left(\left\langle a_{1}\right\rangle\left(B_{1}\right), \ldots,\left\langle a_{n}\right\rangle\left(B_{n}\right)\right)
$$

which is an involution by Proposition 9. Using the results of Section 2.4.2, we get that

$$
\begin{aligned}
& \left(\bmod \left(\varphi_{1}\right), \ldots, \bmod \left(\varphi_{n}\right)\right) \in C_{X} \quad \text { iff } \operatorname{su}\left(\varphi_{1}, \ldots, \varphi_{n}\right)=X \\
& \\
& \quad \text { iff } \bmod \left(\bigwedge_{1 \leq i \leq n}\left(a_{i} \leftrightarrow \varphi_{i}\right)\right)=X \\
& \\
& \text { iff } \bigcap_{1 \leq i \leq n} \bmod \left(a_{i} \leftrightarrow \varphi_{i}\right)=X \\
& \\
& \text { iff } \bigcap_{1 \leq i \leq n}\left\langle a_{i}\right\rangle\left(\bmod \left(\varphi_{i}\right)\right)=X \\
& \\
& \\
& \\
& \\
& \\
& \\
& \text { iff }\left(\left\langle a_{1}\right\rangle\left(\bmod f\left(\bmod \left(\varphi_{1}\right)\right), \ldots, \bmod \left(\varphi_{n}\right)\right) \in Y_{X}\right.
\end{aligned}
$$

Thus $f\left(C_{X}\right)=Y_{X}$ whence $f\left(Y_{X}\right)=f\left(f\left(C_{X}\right)\right)=C_{X}$ and $\left.f\right|_{C_{X}}: C_{X} \rightarrow Y_{X}$ is bijective.
This one-to-one correspondence is important since we will later analyse the precise number of realisations of given model sets. Furthermore, this result shows the role of the conjugation function for characterising two-valued model realisability for general ADFs. We can now adapt this characterisation result to the case of bipolar ADFs. More precisely, we give several necessary and sufficient conditions when a given model set is bipolarly realisable. With this characterisation in hand, we can later show that a specific interpretation set fails the necessary conditions and thus cannot be the model set of any BADF. Below, we denote the set of all supersets of a set $X$ of interpretation sets over $A$ by $X^{\uparrow}=\left\{Y \subseteq 2^{A} \mid X \subseteq Y\right\}$.

Proposition 11. Let $A=\left\{a_{1}, \ldots, a_{n}\right\}$ be a vocabulary and $X \subseteq 2^{A}$ be a set of interpretations. The following are equivalent:

1. $X$ is bipolarly realisable.
2. there exist $Y_{1}, \ldots, Y_{n} \in X^{\uparrow}$ such that:
(a) $\left(\bigcap_{i=1}^{n} Y_{i}\right)=X$, and
(b) for each $1 \leq i \leq n$, the set $\left\langle a_{i}\right\rangle\left(Y_{i}\right)$ is bipolar.
3. there exist $Y_{1}, \ldots, Y_{n} \in X^{\uparrow}$ such that
(a) $\left(\bigcap_{i=1}^{n} Y_{i}\right)=X$, and
(b) for each $1 \leq i, j \leq n$, at least one of:

- for all $M \subseteq A,\left(M \in Y_{i} \leftrightarrow a_{i} \in M\right) \rightarrow\left(M \cup\left\{a_{j}\right\} \in Y_{i} \leftrightarrow a_{i} \in M \cup\left\{a_{j}\right\}\right)$; or
- for all $N \subseteq A,\left(N \in Y_{i} \nrightarrow a_{i} \in N\right) \rightarrow\left(N \cup\left\{a_{j}\right\} \in Y_{i} \nrightarrow a_{i} \in N \cup\left\{a_{j}\right\}\right)$.

Proof. (1) $\Rightarrow$ (2): If $X$ is bipolarly realisable, then there exists a bipolar ADF $D=(A, L, C)$ with $s u(D)=X$. In particular, there exist bipolar Boolean functions $C_{1}, \ldots, C_{n}$ such that $M \in X$ if and only if for all $1 \leq i \leq n$ we find $a_{i} \in M$ iff $C_{i}(M)=\mathbf{t}$. For each $1 \leq i \leq n$ define $Y_{i}=\left\langle a_{i}\right\rangle\left(C_{i}\right)$. By assumption, $\left\langle a_{i}\right\rangle\left(Y_{i}\right)=\left\langle a_{i}\right\rangle\left(\left\langle a_{i}\right\rangle\left(C_{i}\right)\right)=C_{i}$ is bipolar; furthermore $\left(\bigcap_{i=1}^{n} Y_{i}\right)=X$ follows from the above.
$(2) \Rightarrow(3)$ : Let $i \in\{1, \ldots, n\}$ and assume that $\left\langle a_{i}\right\rangle\left(Y_{i}\right)$ is bipolar. This means that for all $a_{j} \in A$, we find that $a_{j}$ is supporting or attacking (or both) in $\left\langle a_{i}\right\rangle\left(Y_{i}\right)$. Now $a_{j}$ is supporting in $\left\langle a_{j}\right\rangle\left(Y_{i}\right)$ iff for all $M \subseteq A$ we find:

$$
\begin{aligned}
M \in\left\langle a_{i}\right\rangle\left(Y_{i}\right) & \rightarrow M \cup\left\{a_{j}\right\} \in\left\langle a_{i}\right\rangle\left(Y_{i}\right), \text { that is, } \\
\left(M \in Y_{i} \leftrightarrow a_{i} \in M\right) & \rightarrow\left(M \cup\left\{a_{j}\right\} \in Y_{i} \leftrightarrow a_{i} \in M \cup\left\{a_{j}\right\}\right)
\end{aligned}
$$

Similarly, $a_{j}$ is attacking in $\left\langle a_{i}\right\rangle\left(Y_{i}\right)$ iff for all $N \subseteq A$ we find:

$$
\begin{aligned}
N \notin\left\langle a_{i}\right\rangle\left(Y_{i}\right) & \rightarrow N \cup\left\{a_{j}\right\} \notin\left\langle a_{i}\right\rangle\left(Y_{i}\right), \text { that is, } \\
\neg\left(N \in Y_{i} \leftrightarrow a_{i} \in N\right) & \rightarrow \neg\left(N \cup\left\{a_{j}\right\} \in Y_{i} \leftrightarrow a_{i} \in N \cup\left\{a_{j}\right\}\right)
\end{aligned}
$$

Thus for all $a_{j} \in A$, we find that at least one of the following:

- for all $M \subseteq A,\left(M \in Y_{i} \leftrightarrow a_{i} \in M\right) \rightarrow\left(M \cup\left\{a_{j}\right\} \in Y_{i} \leftrightarrow a_{i} \in M \cup\left\{a_{j}\right\}\right)$; or
- for all $N \subseteq A,\left(N \in Y_{i} \nleftarrow a_{i} \in N\right) \rightarrow\left(N \cup\left\{a_{j}\right\} \in Y_{i} \nleftarrow a_{i} \in N \cup\left\{a_{j}\right\}\right)$.
$(3) \Rightarrow(1)$ : We construct an ADF $D=(A, L, C)$ as follows: for each $i \in\{1, \ldots, n\}$ we define $C_{i}=\left\langle a_{i}\right\rangle\left(Y_{i}\right)$ and finally set $L=A \times A$. Each $C_{i}$ is bipolar by the equivalences established in the previous proof item, and $s u(D)=X$ follows from the fact that $\left\langle a_{i}\right\rangle\left(C_{i}\right)=\left\langle a_{i}\right\rangle\left(\left\langle a_{i}\right\rangle\left(Y_{i}\right)\right)=Y_{i}$ and the presumption $\left(\bigcap_{i=1}^{n} Y_{i}\right)=X$.

We now apply this characterisation result to show that there is an interpretation set over three atoms that cannot be realised by bipolar ADFs under the model semantics. This is the smallest example in terms of the number of atoms (actually, one of the two smallest examples) - all interpretation sets over a binary vocabulary are bipolarly realisable.

Proposition 12. For vocabulary $A_{3}=\{1,2,3\}$, there is no bipolar ADF that realises $X=\operatorname{EvEN}_{3}=\{\emptyset,\{1,2\},\{1,3\},\{2,3\}\}$.

Proof. Assume to the contrary that $X$ is bipolarly realisable. Then there exist $Y_{1}, Y_{2}, Y_{3} \in X^{\uparrow}$ from Proposition 11. There are $2^{\left|2^{A}\right|-|X|}=2^{8-4}=2^{4}=16$ candidates for each $Y_{i}$, that is, every $Y_{i}$ must be of the form $X \uplus Z$ with

$$
Z \subseteq\{\{1\},\{2\},\{3\},\{1,2,3\}\}=2^{A} \backslash X
$$

For eleven out of those sixteen model set candidates for each $Y_{i}$, the set $\langle i\rangle\left(Y_{i}\right)$ is not bipolar. To show that a model set $\langle i\rangle\left(Y_{i}\right)$ is not bipolar, we provide a statement $j \in A_{3}$ that is neither supporting nor attacking; we say that such a statement is dependent.

1. For $Y_{1}=X$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2\},\{1,3\},\{2\},\{3\}\}$, which is not bipolar since statement 2 is dependent: If 2 was supporting, then $\{3\} \in\langle 1\rangle\left(Y_{1}\right)$ would imply $\{2,3\} \in\langle 1\rangle\left(Y_{1}\right)$; if 2 was attacking, then $\emptyset \notin\langle 1\rangle\left(Y_{1}\right)$ would imply $\{2\} \notin\langle 1\rangle\left(Y_{1}\right)$. For the remaining cases, the justifications for a specific statement being dependent are equally easy to read off the model set; for brevity we just indicate the statements.
2. For $Y_{1}=X \cup\{\{1\}\}$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2\},\{1,3\},\{1\},\{2\},\{3\}\}$, which is not bipolar since statement 2 is dependent.
3. For $Y_{1}=X \cup\{\{2\}\}$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2\},\{1,3\},\{3\}\}$, which is not bipolar since statement 2 is dependent.
4. The case $Y_{1}=X \cup\{\{3\}\}$ is symmetric to the previous one: we get the model set $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2\},\{1,3\},\{2\}\}$, which is not bipolar since statement 3 is dependent.
5. For $Y_{1}=X \cup\{\{1,2,3\}\}$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2,3\},\{1,2\},\{1,3\},\{2\},\{3\}\}$, which is not bipolar since statement 2 is dependent.
6. For $Y_{1}=X \cup\{\{1\},\{2\}\}$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2\},\{1,3\},\{1\},\{3\}\}$, which is not bipolar since statement 3 is dependent.
7. The case $Y_{1}=X \cup\{\{1\},\{3\}\}$ is again symmetric to the previous one.
8. For $Y_{1}=X \cup\{\{2\},\{3\}\}$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2\},\{1,3\}\}$, which is not bipolar since statement 2 is dependent.
9. For $Y_{1}=X \cup\{\{1\},\{1,2,3\}\}$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2,3\},\{1,2\},\{1,3\},\{1\},\{2\},\{3\}\}$, which is not bipolar since statement 2 is dependent.
10. For $Y_{1}=X \cup\{\{2\},\{1,2,3\}\}$, we get $\langle 1\rangle\left(Y_{1}\right)=\{\{1,2,3\},\{1,2\},\{1,3\},\{3\}\}$, which is not bipolar since statement 2 is dependent.
11. $Y_{1}=X \cup\{\{3\},\{1,2,3\}\}$ is again symmetric to the previous case.

There remains a set $C$ of five candidates (due to symmetry they are the same for each $i$ ):

$$
\begin{aligned}
C=\{ & X \uplus\{\{1\},\{2\},\{3\}\}, \\
& X \uplus\{\{1\},\{2\},\{1,2,3\}\}, \\
& X \uplus\{\{1\},\{3\},\{1,2,3\}\}, \\
& X \uplus\{\{2\},\{3\},\{1,2,3\}\}, \\
& X \uplus\{\{1\},\{2\},\{3\},\{1,2,3\}\}\}
\end{aligned}
$$

Basically, the candidates are those where at least three out of the four interpretations in $D=\{\{1\},\{2\},\{3\},\{1,2,3\}\}$ are contained in addition to those already in $X$. Now clearly by the assumption that the $Y_{i}$ realise $X$ we have $Y_{1}, Y_{2}, Y_{3} \in C$. But then there is some $M \in D$ with $M \in Y_{i}$ for all $1 \leq i \leq 3$ and thus $M \in\left(\bigcap_{i=1}^{3} Y_{i}\right)=X$. However, $D \cap X=\emptyset$. Contradiction. Thus such $Y_{i}$ do not exist and $X$ is not bipolarly realisable.

As the only other interpretation set over $A_{3}$ that is not bipolarly realisable, we found the complement of $\mathrm{EvEn}_{3}$ above, the Parity language over three atoms.

Proposition 13. For vocabulary $A_{3}=\{1,2,3\}$, there is no bipolar ADF that realises Parity $_{3}=\{\{1\},\{2\},\{3\},\{1,2,3\}\}$.

Together with the straightforward statement of fact that $\operatorname{EvEN}_{3}$ can be realised by a non-bipolar ADF, Proposition 12 leads to the next result.
Theorem 14. $B A D F^{s u}<_{e} A D F^{s u}$
Proof. Model set EvEn 3 from Proposition 12 is realisable under model semantics by ADF $D_{\text {Even }_{3}}$ with acceptance conditions

$$
\varphi_{1}=(2 \leftrightarrow 3), \quad \varphi_{2}=(1 \leftrightarrow 3), \quad \varphi_{3}=(1 \leftrightarrow 2)
$$

However, there is no bipolar ADF realising $\mathrm{Even}_{3}$, as is witnessed by Proposition 12.
Another consequence of our characterisation of two-valued model realisability in Proposition 10 is that we can get a precise number of distinct realisations of a given model set. This is significant in that it further illustrates the rather intricate difficulty underlying bipolar non-realisability: we cannot necessarily use the model set $\operatorname{EvEN}_{3}$ above to determine a single reason for bipolar non-realisability, that is, a single link $(b, a)$ that is neither supporting nor attacking in all realisations. Rather, the culprit(s) might be different in each realisation, and to show bipolar non-realisability, we have to prove that for all realisations, there necessarily exists some reason for non-bipolarity. And the number of different ADF realisations of a given model set $X$ can be considerable. ${ }^{8}$

Proposition 15. Let $A$ be a vocabulary with $|A|=n$, and $X \subseteq 2^{A}$ an interpretation set with $\left|2^{A} \backslash X\right|=m$. The number of distinct ADFs $D$ with su $(D)=X$ is

$$
r(n, m)=\left(2^{n}-1\right)^{m}
$$

Proof. According to Proposition 10, each realisation of $X$ can be characterised by a tuple $\left(Y_{1}, \ldots, Y_{n}\right) \in\left(X^{\uparrow}\right)^{n}$ with $X=\bigcap_{i=1}^{n} Y_{i}$. Since $\left|X^{\uparrow}\right|=2^{m}$, there are $\left(2^{m}\right)^{n}$ such tuples. However, towards $r(n, m)$, this wrongly counts all tuples $\left(Y_{1}, \ldots, Y_{n}\right)$ with $\left(\bigcap_{i=1}^{n} Y_{i}\right) \supsetneq X$, that is, $\left|\left(\bigcap_{i=1}^{n} Y_{i}\right) \backslash X\right|>0$ (at least once); it remains to subtract them. For any $i \in\{1, \ldots, n\}$, we can overestimate the number of tuples $\left(Y_{1}, \ldots, Y_{n}\right) \in\left(X^{\uparrow}\right)^{n}$ such that $\left|\left(\bigcap_{i=1}^{n} Y_{i}\right) \backslash X\right| \geq i$ by the expression

$$
\begin{equation*}
q(n, m, i)=\binom{m}{i}\left(2^{m-i}\right)^{n} \tag{3}
\end{equation*}
$$

8. When counting ADFs over $A$, we do not take into account different link relations, but take $L=A \times A$ and only count different acceptance functions, through which redundant links can be modelled.

This is seen as follows: Let $I \subseteq\left(2^{A} \backslash X\right)$ be a fixed $i$-element set. (Intuitively, the inter-pretation-set $X \cup I$ contains exactly $i$ interpretations too many.) There are ( $\left.\begin{array}{c}m \\ i\end{array}\right)$ such sets. For each such $I$, we have $\left|I^{\uparrow}\right|=2^{m-i}$. Thus there are $\left(2^{m-i}\right)^{n}$ possible ways to choose $n$ elements (the $Y_{1}, \ldots, Y_{n}$ ) out of $I^{\uparrow}$. No matter how the $Y_{j}$ are chosen, their intersection contains $I$ and thus has at least $i$ elements too many. However, all sets that have at least $i+1$ elements too many are counted twice and have to be subtracted. If we subtract $q(n, m, i+1)$, then we have not counted the sets that have at least $i+2$ elements too many and have to add $q(n, m, i+2)$, etc. Hence by the inclusion-exclusion principle, the number of tuples $\left(Y_{1}, \ldots, Y_{n}\right) \in\left(X^{\uparrow}\right)^{n}$ with $\bigcap_{i=1}^{n} Y_{i}=X$ is given by

$$
\begin{array}{rlr}
r(n, m) & =q(n, m, 0)-q(n, m, 1)+q(n, m, 2)-\ldots \pm q(n, m, m) \\
& =\sum_{i=0}^{m}(-1)^{i} q(n, m, i) & \\
& =\sum_{i=0}^{m}(-1)^{i}\binom{m}{i}\left(2^{m-i}\right)^{n} & \\
& =\sum_{i=0}^{m}\binom{m}{i}\left(2^{n}\right)^{m-i}(-1)^{i} & \\
& =\left(2^{n}-1\right)^{m} & \text { (reordering factors) }
\end{array}
$$

So the main contributing factor is the number $m$ of interpretations that are excluded from the desired model set $X$. For Proposition 12, for instance, there are $\left(2^{3}-1\right)^{4}=7^{4}=2401$ ADFs with the model set $\mathrm{EvEn}_{3}$. According to Theorem 14, none of them is bipolar. Obviously, the maximal number of realisations is achieved by $X=\emptyset$ whence $r\left(n, 2^{n}\right)=\left(2^{n}-1\right)^{2^{n}}$. On the other hand, the model set $X=2^{A}$ has exactly one realisation, $r(n, 0)=1$. Note that the number of (syntactically distinct) realisations for the other universally expressive formalisms, logic programs and propositional logic, is unbounded in general since we can add an arbitrary number of tautologies.

We finally show a reduction of the problem of bipolar realisability to propositional satisfiability. This approaches the problem from another angle (a possible implementation deciding bipolar realisability using a SAT solver), and provides the proof of Theorem 3 by Strass (2015b), which was not contained in that work.

For a given vocabulary $A$ and set $X \subseteq 2^{A}$ be a set of interpretations, it is our aim to construct a propositional formula $\phi_{X}$ that is satisfiable if and only if $X$ is bipolarly realisable. The propositional signature we use is the following: For each $a \in A$ and $M \subseteq A$, there is a propositional variable $p_{a}^{M}$ that expresses whether $C_{a}(M)=\mathbf{t}$. This allows to encode all possible acceptance conditions for the statements in $A$. To enforce bipolarity, we use additional variables to model supporting and attacking links: for all $a, b \in A$, there is a variable $p_{\text {sup }}^{a, b}$ saying that $a$ supports $b$, and a variable $p_{a t t}^{a, b}$ saying that $a$ attacks $b$. So the vocabulary of $\phi_{X}$ is given by

$$
P=\left\{p_{a}^{M}, p_{\text {sup }}^{a, b}, p_{a t t}^{a, b} \mid M \subseteq A, a \in A, b \in A\right\}
$$

To guarantee the desired set of models, we constrain the acceptance conditions as dictated by $X$ : For any desired set $M$ and statement $a$, the containment of $a$ in $M$ must correspond
exactly to whether $C_{a}(M)=\mathbf{t}$; this is encoded in $\phi_{X}^{\in}$. Conversely, for any undesired set $M$ and statement $a$, there must not be any such correspondence, which $\phi_{X}^{\neq}$expresses. To enforce bipolarity, we state that each link must be supporting or attacking. To model the meaning of support and attack, we encode all ground instances of their definitions.

Definition 6. Let $A$ be a vocabulary and $X \subseteq 2^{A}$ be a set of interpretations. Define the following propositional formulas:

$$
\begin{aligned}
\phi_{X}^{\mathrm{BADF}} & =\phi_{X}^{\in} \wedge \phi_{X}^{\notin} \wedge \phi_{\text {bipolar }} \\
\phi_{X}^{\in} & =\bigwedge_{M \in X}\left(\bigwedge_{a \in M} p_{a}^{M} \wedge \bigwedge_{a \in A \backslash M} \neg p_{a}^{M}\right) \\
\phi_{X}^{\notin} & =\bigwedge_{M \subseteq A, M \notin X}\left(\bigvee_{a \in M} \neg p_{a}^{M} \vee \bigvee_{a \in A \backslash M} p_{a}^{M}\right) \\
\phi_{\text {bipolar }} & =\bigwedge_{a, b \in A}\left(\left(p_{\text {sup }}^{a, b} \vee p_{a t t}^{a, b}\right) \wedge \phi_{\text {sup }}^{a, b} \wedge \phi_{a t t}^{a, b}\right) \\
\phi_{\text {sup }}^{a, b} & =p_{\text {sup }}^{a, b} \rightarrow \bigwedge_{M \subseteq A}\left(p_{b}^{M} \rightarrow p_{b}^{M \cup\{a\}}\right) \\
\phi_{a t t}^{a, b} & =p_{a t t}^{a, b} \rightarrow \bigwedge_{M \subseteq A}\left(p_{b}^{M \cup\{a\}} \rightarrow p_{b}^{M}\right)
\end{aligned} \quad(a, b \in A)
$$

The corresponding result shows the reduction to be correct.
Theorem 16. Let $A$ be a vocabulary and $X \subseteq 2^{A}$ be a set of interpretations. $X$ is bipolarly realisable if and only if $\phi_{X}^{\mathrm{BADF}}$ is satisfiable.

Proof. "if": Let $I \subseteq P$ be a model for $\phi_{X}$. For each $a \in A$, we define an acceptance condition as follows: for $M \subseteq A$, set $C_{a}(M)=\mathbf{t}$ iff $p_{a}^{M} \in I$. It is easy to see that $\phi_{b i p o l a r}$ guarantees that these acceptance conditions are all bipolar. The ADF is now given by $D_{X}^{s u}=(A, A \times A, C)$. It remains to show that any $M \subseteq A$ is a model of $D_{X}^{s u}$ if and only if $M \in X$.
"if": Let $M \in X$. We have to show that $M$ is a model of $D_{X}^{s u}$. Consider any $a \in A$.

1. $a \in M$. Since $I$ is a model of $\phi_{X}^{\in}$, we have $p_{a}^{M} \in I$ and thus by definition $C_{a}(M)=\mathbf{t}$.
2. $a \in A \backslash M$. Since $I$ is a model of $\phi_{X}^{\in}$, we have $p_{a}^{M} \notin I$ and thus by definition $C_{a}(M)=\mathbf{f}$.
"only if": Let $M \notin X$. Since $I$ is a model of $\phi_{X}^{\neq}$, there is an $a \in M$ such that $C_{a}(M)=\mathbf{f}$ or an $a \notin M$ such that $C_{a}(M)=\mathbf{t}$. In any case, $M$ is not a model of $D_{X}^{s u}$.
"only if": Let $D$ be a bipolar ADF with $s u(D)=X$. We use $D$ to define a model $I$ for $\phi_{X}$. First, for $M \subseteq A$ and $a \in A$, set $p_{a}^{M} \in I$ iff $C_{a}(M)=\mathbf{t}$. Since $D$ is bipolar, each link is supporting or attacking and for all $a, b \in A$ we can find a valuation for $p_{s u p}^{a, b}$ and $p_{a t t}^{a, b}$. It remains to show that $I$ is a model for $\phi_{X}$.
3. $I$ is a model for $\phi_{X}^{\in}$ : Since $D$ realises $X$, each $M \in X$ is a model of $D$ and thus for all $a \in A$ we have $C_{a}(M)=\mathbf{t}$ iff $a \in M$.
4. $I$ is a model for $\phi_{X}^{\neq}$: Since $D$ realises $X$, each $M \subseteq A$ with $M \notin X$ is not a model of $D$. Thus for each such $M$, there is an $a \in A$ witnessing that $M$ is not a model of $D:(1) a \in M$ and $C_{a}(M)=\mathbf{f}$, or (2) $a \notin M$ and $C_{a}(M)=\mathbf{t}$.
5. $I$ is a model for $\phi_{\text {bipolar }}$ : straightforward since $D$ is bipolar by assumption.

Remarkably, the decision procedure does not only give an answer, but in the case of a positive answer we can read off the BADF realisation from the satisfying evaluation of the constructed formula. We illustrate the construction with an example seen earlier.

Example 2. Consider $A_{3}=\{1,2,3\}$ and the model set $\operatorname{EvEN}_{3}=\{\emptyset,\{1,2\},\{1,3\},\{2,3\}\}$. The construction of Theorem 16 yields these formulas:

$$
\begin{aligned}
& \phi_{\mathrm{EVEN}_{3}}^{\in}=\neg p_{1}^{\emptyset} \wedge \neg p_{2}^{\emptyset} \wedge \neg p_{3}^{\emptyset} \wedge \\
& p_{1}^{\{1,2\}} \wedge p_{2}^{\{1,2\}} \wedge \neg p_{3}^{\{1,2\}} \wedge \\
& p_{1}^{\{1,3\}} \wedge \neg p_{2}^{\{1,3\}} \wedge p_{3}^{\{1,3\}} \wedge \\
& \neg p_{1}^{\{2,3\}} \wedge p_{2}^{\{2,3\}} \wedge p_{3}^{\{2,3\}}
\end{aligned}
$$

$$
\begin{aligned}
\phi_{\mathrm{EVEN}_{3}}= & \left(\neg p_{1}^{\{1\}} \vee p_{2}^{\{1\}} \vee p_{3}^{\{1\}}\right) \wedge \\
& \left(p_{1}^{\{2\}} \vee \neg p_{2}^{\{2\}} \vee p_{3}^{\{2\}}\right) \wedge \\
& \left(p_{1}^{\{3\}} \vee p_{2}^{\{3\}} \vee \neg p_{3}^{\{3\}}\right) \wedge \\
& \left(\neg p_{1}^{\{1,2,3\}} \vee \neg p_{2}^{\{1,2,3\}} \vee \neg p_{3}^{\{1,2,3\}}\right)
\end{aligned}
$$

The remaining formulas about bipolarity are independent of $\operatorname{EVEN}_{3}$, we do not show them here. We have implemented the translation of the proof of Theorem 16 and used the solver clasp (Gebser, Kaminski, Kaufmann, Ostrowski, Schaub, \& Schneider, 2011) to verify that $\phi_{\text {Even }_{3}}$ is unsatisfiable.

### 3.1.4 BADF vs. LP

Earlier, we used the language Parity to show that propositional logic is (and thus by $\mathrm{PL} \cong{ }_{s} \mathrm{ADF}^{s u}$ general ADFs are) exponentially more succinct than normal logic programs (under supported models). However, for bipolar ADFs, by Proposition 13 there is no BADF $D$ over $A_{3}=\{1,2,3\}$ with model set $s u(D)=$ Parity $_{3}=\{\{1\},\{2\},\{3\},\{1,2,3\}\}$, that is, BADFs cannot even express Parity. Fortunately, the Majority language does the trick in this case.

Theorem 17. $B A D F^{s u} \not \pm_{s} L P^{s u}$
Proof. We show that the language Majority can be polynomially expressed by BADF ${ }^{\text {su }}$, but not by LP ${ }^{s u}$. The latter fact follows from Majority $\notin \mathrm{AC}^{0}$ (Jukna, 2012) and Proposition 6. We show the first part by constructing a series of BADFs $D_{n}$ over $A_{n}=\left\{a_{1}, \ldots, a_{n}\right\}$ $(n \in \mathbb{N}, n \geq 1)$ such that $s u\left(D_{n}\right)=$ MAJority $_{n}$. We use results of (Friedman, 1986; Boppana, 1986), who show that for all positive $n \in \mathbb{N}$ and $k \leq n$, the language $\operatorname{Threshold}_{n, k}$ has negation-free propositional formulas $\Phi_{n, k}^{\text {Threshold }}$ of polynomial size $s$, where we use the bound of Boppana, $s \in O\left(k^{4.27} n \log n\right)$. Define $D_{1}$ by $\varphi_{a_{1}}=\top$, and for $n \geq 2$ set $k=\left\lceil\frac{n}{2}\right\rceil$ and for $1 \leq i \leq n$,

$$
\varphi_{a_{i}}=a_{i} \vee \neg \Phi_{n-1, k}^{\mathrm{THRESHOLD}}\left(a_{1}, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{n}\right)
$$

Intuitively, the formula $\varphi_{a_{i}}$ checks whether the remaining variables could achieve a majority without $a_{i}$. If so, then $a_{i}$ can be set arbitrarily; otherwise, $a_{i}$ must be set to true. Clearly the Boolean function computed by $\varphi_{a_{i}}$ is bipolar, since $a_{i}$ is supporting and all other parents are attacking. For the size of $D_{n}$, we observe that

$$
\left\|D_{n}\right\| \in O\left(n\left\|\Phi_{n-1, k}^{\text {ThreshoLD }}\right\|\right)
$$

whence the overall size is polynomial. It remains to show that $s u\left(D_{n}\right)=$ Majority $_{n}$.
" $\supseteq$ ": Let $M \in$ MAJORITY $_{n}$. We have to show $M \in \operatorname{su}\left(D_{n}\right)$, that is, $a \in M$ iff $M \models \varphi_{a}$ for all $a \in A_{n}$. For $a \in M$, it is immediate that $M \models \varphi_{a}$, so let $a_{j} \notin M$ for some $j \in\{1, \ldots, n\}$. We have to show $M \not \vDash \varphi_{a_{j}}$. Since $M \in$ Majority $_{n}$, we have $|M|=m$ for $k=\left\lceil\frac{n}{2}\right\rceil \leq m \leq n-1$ and $M \in \operatorname{THRESHOLD}_{n-1, k}$, that is, we have

$$
M \models \Phi_{n-1, k}^{\mathrm{ThRESHOLD}}\left(a_{1}, \ldots, a_{j-1}, a_{j+1}, \ldots, a_{n}\right)
$$

Together with $M \not \vDash a_{j}$, it follows that $M \not \vDash \varphi_{a_{j}}$.
" $\subseteq$ ": Let $M \notin$ Majority $_{n}$. Then $|M|=m$ for $0 \leq m<\left\lceil\frac{n}{2}\right\rceil=k$. In particular, there is some $a_{j} \in A_{n} \backslash M$. Now $m<k$ implies that there is no $N \in \operatorname{THRESHOLD}_{n-1, k}$ with $|N|=m=|M|$. Thus $M \not \vDash \Phi_{n-1, k}^{\text {Threshold }}\left(a_{1}, \ldots, a_{j-1}, a_{j+1}, \ldots, a_{n}\right)$ whence it follows that $M \models \varphi_{a_{j}}$. Together with $M \not \vDash a_{j}$ we conclude that $M \notin s u\left(D_{n}\right)$.

Since every BADF is an ADF of the same size, we get:
Corollary 18. $A D F^{s u} \not_{s} L P^{s u}$
In combination with the translation from logic programs to ADFs (implying the relation $\mathrm{LP}^{s u} \leq_{s} \mathrm{ADF}^{s u}$ ), this means that also ADFs are strictly more succinct than logic programs.
Corollary 19. $L P^{s u}<_{s} A D F^{s u}$

### 3.1.5 BADF vs. AF

It is comparably easy to show that BADF models are strictly more expressive than AFs, since sets of supported models of bipolar ADFs do not have the antichain property.

Proposition 20. $A F<_{e} B A D F^{s u}$
Proof. Consider vocabulary $A=\{a\}$ and $\operatorname{BADF} D=\left(A,\{(a, a)\},\left\{\varphi_{a}\right\}\right)$ with $\varphi_{a}=a$. It is straightforward to check that its model set is $s u(D)=\{\emptyset,\{a\}\}$. Since model sets of AFs under stable extension semantics satisfy the antichain property, there is no equivalent AF over $A$.

This yields the following overall relationships:

$$
\mathrm{AF}<_{e} \mathrm{BADF}^{s u}<_{e} \mathrm{ADF}^{s u} \cong_{e} \mathrm{LP}^{s u} \cong_{e} \mathrm{PL}
$$

For a concise overview of relative succinctness, we present the results and open problems at a glance in Table 2 below. ${ }^{9}$

[^3]|  | $\mathrm{BADF}^{s u}$ | $\mathrm{ADF}^{s u}$ | LP |  |
| :--- | :---: | :---: | :---: | :---: |
| $\mathrm{PADF}^{s u}$ | $=$ | $\leq_{s}$ | $\mathbf{Z}_{s}$ | $\leq_{s}$ |
| $\mathrm{ADF}^{s u}$ | $?$ | $=$ | $\mathbf{Z}_{s}$ | $\cong_{s}$ |
| $\mathrm{LP}^{s u}$ | $?$ | $<_{s}$ | $=$ | $<_{s}$ |
| PL | $?$ | $\cong_{s}$ | $\mathbf{Z}_{s}$ | $=$ |

Table 2: Relative succinctness results for (bipolar) ADFs under the model semantics, normal logic programs under the supported semantics, and classical propositional logic. An entry $\circ$ in row $\mathcal{F}_{1}$ and column $\mathcal{F}_{2}$ means $\mathcal{F}_{1} \circ \mathcal{F}_{2}$.

### 3.2 Stable Semantics

As before, we recall the current state of knowledge:

$$
\mathrm{AF} \leq_{e} \mathrm{BADF}^{s t} \leq_{e} \mathrm{ADF}^{s t}<_{e} \mathrm{PL} \text { and } \mathrm{AF} \leq_{e} \mathrm{LP}^{s t}<_{e} \mathrm{PL}
$$

We first show that BADFs are strictly more expressive than AFs.
Proposition 21. $A F<_{e} B A D F^{s t}$
Proof. Consider the set $X_{2}=\{\{a, b\},\{a, c\},\{b, c\}\}$ of desired models. Dunne et al. (2015) proved that $X_{2}$ is not realisable with stable AF semantics. However, the model set $X_{2}$ is realisable with BADF $D_{X_{2}}$ under stable semantics:

$$
\varphi_{a}=\neg b \vee \neg c, \quad \varphi_{b}=\neg a \vee \neg c, \quad \varphi_{c}=\neg a \vee \neg b
$$

Let us exemplarily show that $M=\{a, b\}$ is a stable model (the other cases are completely symmetric): The reduct $D^{M}$ is characterised by the two acceptance formulas $\varphi_{a}=\neg b \vee \neg \perp$ and $\varphi_{b}=\neg a \vee \neg \perp$. We then easily find that $\Gamma_{D^{M}}(\emptyset, \emptyset)=(M, \emptyset)=\Gamma_{D^{M}}(M, \emptyset)$.

Intuitively, the argument for AF non-realisability of $X_{2}$ is as follows: Since $a$ and $b$ occur in an extension together, there can be no attack between them. The same holds for the pairs $a, c$ and $b, c$. But then the set $\{a, b, c\}$ is conflict-free and thus there must be a stable extension containing all three arguments, which is not allowed by $X_{2}$. The reason is AFs' restriction to individual attack, as set attack (also called joint or collective attack) suffices to realise $X_{2}$ as seen above.

The construction that we used in the proof above to realize $X_{2}$ comes from the work of Eiter, Fink, Pührer, Tompits, and Woltran (2013) in logic programming, and can be generalised to realise any non-empty model set satisfying the antichain property.

Definition 7. Let $X \subseteq 2^{A}$. Define the following BADF $D_{X}^{s t}=(A, L, C)$ where $C_{a}$ for $a \in A$ is given by

$$
\varphi_{a}=\bigvee_{M \in X, a \in M}\left(\bigwedge_{b \in A \backslash M} \neg b\right)
$$

and thus $L=\{(b, a) \mid M \in X, a \in M, b \in A \backslash M\}$.

The next result shows that the construction indeed works.
Theorem 22. Let $X$ with $\emptyset \neq X \subseteq 2^{A}$ be a $\subseteq$-antichain. We find that $\operatorname{st}\left(D_{X}^{s t}\right)=X$.
Proof. Let $M \subseteq A$.
" $\subseteq$ ": Let $M \notin X$. We show that $M \notin s u\left(D_{X}^{s t}\right) \supseteq s t\left(D_{X}^{s t}\right)$; we use a case distinction.

1. There is an $N \in X$ with $M \subsetneq N$. Then there is an $a \in N \backslash M$. Consider its acceptance formula $\varphi_{a}$. Since $a \in N$ and $N \in X$, the formula $\varphi_{a}$ has a disjunct $\psi_{a, N}=\bigwedge_{b \in A \backslash N} \neg b$. Now $M \subseteq N$ implies $A \backslash N \subseteq A \backslash M$ and $M$ is a model for $\psi_{a, N}$. Thus $M$ is a model for $\varphi_{a}$ although $a \notin M$, hence $M \notin s u\left(D_{X}^{s t}\right)$.
2. For all $N \in X$, we have $M \nsubseteq N$. Then $X \neq \emptyset$ implies $M \neq \emptyset$, so let $a \in M$. For each $N \in X$ with $a \in N$, the acceptance formula $\varphi_{a}$ contains a disjunct $\psi_{a, N}=\bigwedge_{b \in A \backslash N} \neg b$. By assumption, for each $N \in X$ there is a $b_{N} \in M \backslash N$. Clearly $b_{N} \in A \backslash N$ and $b_{N}$ is evaluated to true by $M$. Hence for each $N \in X$ with $a \in N$, the disjunct $\psi_{a, N}$ is evaluated to false by $M$. Thus $\varphi_{a}$ is false under $M$ and $M \notin s u\left(D_{X}^{s t}\right)$.
" $\supseteq$ ": Let $M \in X$. We first show that $M \in s u\left(D_{X}^{s t}\right)$, that is: for all $a \in A$, we find $a \in M$ iff $M$ is a model for $\varphi_{a}$.
3. Let $a \in M$. By construction, we have that $\varphi_{a}$ in $D_{X}^{s t}$ contains a disjunct of the form $\psi_{a, M}=\bigwedge_{b \in A \backslash M} \neg b$. According to the interpretation $M$, all such $b \in A \backslash M$ are false and thus $\psi_{a, M}$ is true whence $\varphi_{a}$ is true.
4. Let $a \in A \backslash M$ and consider its acceptance formula $\varphi_{a}$. Assume to the contrary that $M$ is a model for $\varphi_{a}$. Then there is some $N \in X$ with $a \in N$ such that $M$ is a model for $\psi_{a, N}=\bigwedge_{b \in A \backslash N} \neg b$, that is, $A \backslash N \subseteq A \backslash M$. Hence $M \subseteq N$; and, since $a \in N \backslash M$, even $M \subsetneq N$, whence $X$ is not a $\subseteq$-antichain. Contradiction. Thus $M$ is no model for $\varphi_{a}$.

Now consider the reduct $D^{M}$ of $D_{X}^{s t}$ with respect to $M$. There, $\varphi_{a}^{M}$ contains the disjunct $\psi_{a, M}^{M}=\psi_{a, M}[b / \perp: b \notin M]$ where all $b \in A \backslash M$ have been replaced by false, whence $\psi_{a, M}^{M}=\neg \perp \wedge \ldots \wedge \neg \perp$ and $\varphi_{a}^{M}$ is equivalent to true. Thus each $a \in M$ is true in the least fixpoint of $\Gamma_{D^{M}}$ and thus $M \in \operatorname{st}\left(D_{X}^{s t}\right)$.

The restriction to non-empty model sets is immaterial for relative expressiveness, since we can use the construction of Theorem 2 and the fact that $s t(D) \subseteq s u(D)$ for any ADF $D$ to realize the empty model set. As the stable model semantics for ADFs and logic programs both have the antichain property, we get:

Corollary 23. $A D F^{s t} \leq_{e} B A D F^{s t}$ and $L P^{s t} \leq_{e} B A D F^{s t}$
This leads to the following overall relationships:

$$
\mathrm{AF}<_{e} \mathrm{BADF}^{s t} \cong_{e} \mathrm{ADF}^{s t} \cong_{e} \mathrm{LP}^{s t}<_{e} \mathrm{PL}
$$

We remark that the antichain property provides a characterisation of realisability with the stable semantics; that is, a model set is stable-realisable iff it is a $\subseteq$-antichain.

### 3.3 Supported vs. Stable Semantics

Now we put the supported and stable pictures together. From the proof of Theorem 22, we can read off that for the canonical realisation $D_{X}^{s t}$ of an antichain $X$, the supported and stable semantics coincide, that is, $s u\left(D_{X}^{s t}\right)=s t\left(D_{X}^{s t}\right)=X$. With this observation, also bipolar ADFs under the supported semantics can realize any antichain, and we have this:

Proposition 24. $B A D F^{s t} \leq_{e} B A D F^{s u}$
As we have seen in Proposition 20, there are bipolar ADFs with supported-model sets that are not antichains. We get:

Corollary 25. $B A D F^{s t}<_{e} B A D F^{s u}$

This result allows us to close the last gap and put together the big picture on relative expressiveness in Figure 2 below.


Figure 2: The expressiveness hierarchy. Expressiveness strictly increases from bottom to top. $\mathcal{F}^{\sigma}$ denotes formalism $\mathcal{F}$ under semantics $\sigma$, where "su" is the supported and "st" the stable model semantics; formalisms are among AFs (argumentation frameworks), ADFs (abstract dialectical frameworks), BADFs (bipolar ADFs), LPs (normal logic programs) and PL (propositional logic).

## 4. Allowing Vocabulary Expansion

Up to here, we only considered compact realisations, that do not introduce new vocabulary elements. In this section, we allow the introduction of a small number of new atoms/arguments/statements. More precisely, small means the number is linear in the size of the source knowledge base (representing the model set that we wish to realize in a target language). For the purpose of realisability, the new vocabulary elements are projected out of the resulting models.

As it turns out, adding additional arguments already makes AFs universally expressive (under projection). More technically, we will now show that for each propositional formula $\varphi$ over vocabulary $A$, there exists an $\mathrm{AF} F_{\varphi}$ over an expanded vocabulary $A \cup A_{\varphi}$ such that the models of $\varphi$ and the stable extensions of $F_{\varphi}$ correspond one-to-one. Roughly, this is possible since AFs can be regarded as a syntactic variant of classical propositional logic that has as its only connective the logical NOR " $\downarrow$ " (Gabbay, 2011; Brewka et al., 2011). Using this connective, negation is expressed by $\neg \varphi=\varphi \downarrow \varphi$ and disjunction by
$\varphi \vee \psi=\neg(\varphi \downarrow \psi)=(\varphi \downarrow \psi) \downarrow(\varphi \downarrow \psi)$. These equivalences can be used to translate arbitrary propositional formulas (over $\neg, \wedge, \vee$ ) into the syntactical $\downarrow$-fragment; to guarantee that the size increase is at most linear, we introduce names $a_{\psi}$ for subformulas $\psi$ (Tseitin, 1968). The next definition combines all of these ideas.

Definition 8. Let $\varphi$ be a formula using $\neg, \wedge, \vee$ over vocabulary $A$. Define the sets $A_{\varphi}$ and $R_{\varphi}$ inductively as follows:

$$
\begin{aligned}
& A_{\top}=\left\{a_{\top}\right\} \\
& R_{\top}=\emptyset \\
& A_{\perp}=\left\{a_{\perp}\right\} \\
& R_{\perp}=\left\{\left(a_{\perp}, a_{\perp}\right)\right\} \\
& A_{p}=\left\{p, a_{\neg p}\right\} \text { for } p \in A \\
& R_{p}=\left\{\left(p, a_{\neg p}\right),\left(a_{\neg p}, p\right)\right\} \text { for } p \in A \\
& A_{\neg \xi}=\left\{a_{\neg \xi}\right\} \cup A_{\xi} \quad R_{\neg \xi}=\left\{\left(a_{\xi}, a_{\neg \xi}\right)\right\} \cup R_{\xi} \\
& A_{\zeta \wedge \xi}=\left\{a_{\zeta \wedge \xi}, a_{\neg \zeta}, a_{\neg \xi}\right\} \cup A_{\neg \zeta} \cup A_{\neg \xi} \quad R_{\zeta \wedge \xi}=\left\{\left(a_{\neg \zeta}, a_{\zeta \wedge \xi}\right),\left(a_{\neg \xi}, a_{\zeta \wedge \xi}\right)\right\} \cup R_{\neg \zeta} \cup R_{\neg \xi} \\
& A_{\zeta \vee \xi}=\left\{a_{\zeta \vee \xi}, a_{\zeta \downarrow \xi}\right\} \cup A_{\zeta} \cup A_{\xi} \quad R_{\zeta \vee \xi}=\left\{\left(a_{\zeta \downarrow \xi}, a_{\zeta \vee \xi}\right),\left(a_{\zeta}, a_{\zeta \downarrow \xi}\right),\left(a_{\zeta}, a_{\zeta \downarrow \xi}\right)\right\} \cup R_{\zeta} \cup R_{\xi}
\end{aligned}
$$

The $A F$ associated to $\varphi$ is given by $F_{\varphi}=\left(A_{\varphi} \cup A_{\perp}, R_{\varphi} \cup\left\{\left(a_{\varphi}, a_{\perp}\right)\right\} \cup R_{\perp}\right)$.
The argument $a_{\top}$ is unattacked and thus part of every stable extension (is true in every interpretation); the argument $a_{\perp}$ attacks itself and thus cannot be part of any stable extension (is false in every interpretation). The mutually attacking arguments $p$ and $a_{\neg p}$ for $p \in A$ serve to "guess" a valuation of $A$, while $a_{\varphi}$ and $a_{\perp}$ guarantee that only (and all) valuations that are models of $\varphi$ can lead to stable extensions of $F_{\varphi}$ : intuitively, $a_{\perp}$ must be attacked, and the only candidate to do so is $a_{\varphi}$. The arguments and attacks for the Boolean connectives express their usual truth-theoretic semantics, as our first technical result for this translation shows.

Lemma 26. Let $\varphi$ be a formula over vocabulary $A$ and $F_{\varphi}$ its associated AF. For each stable extension $M$ of $F_{\varphi}$ and $a_{\zeta}, a_{\xi} \in A_{\varphi}$, we have:

- $a_{\neg \xi} \in M$ iff $a_{\xi} \notin M$;
- $a_{\zeta \wedge \xi} \in M$ iff both $a_{\zeta} \in M$ and $a_{\xi} \in M$;
- $a_{\zeta \vee \xi} \in M$ iff one of $a_{\zeta} \in M$ or $a_{\xi} \in M$;
- $a_{\zeta \downarrow \xi} \in M$ iff neither $a_{\zeta} \in M$ nor $a_{\xi} \in M$.

Proof. - By definition, the only attacker of an argument of the form $a_{\neg \xi}$ is the argument $a_{\xi}$. Thus $a_{\xi} \in M$ iff $a_{\neg \xi} \notin M$.

- The only attackers of $a_{\zeta \wedge \xi}$ are the arguments $a_{\neg \zeta}$ and $a_{\neg \xi}$. By the case above, we have $a_{\neg \zeta} \in M$ iff $a_{\zeta} \notin M$, and $a_{\neg \xi} \in M$ iff $a_{\xi} \notin M$. Consequently, $a_{\zeta} \in M$ and $a_{\xi} \in M$ iff $a_{\neg \zeta} \notin M$ and $a_{\neg \xi} \notin M$ iff $a_{\zeta \wedge \xi} \in M$.
- The only attacker of $a_{\zeta \vee \xi}$ is the argument $a_{\zeta \downarrow \xi}$. Similarly to the previous cases, we can show that $a_{\zeta \downarrow \xi} \in M$ iff $a_{\zeta} \notin M$ and $a_{\xi} \notin M$, and that $a_{\zeta \vee \xi} \in M$ iff $a_{\zeta \downarrow \xi} \notin M$. In combination, $a_{\zeta \vee \xi} \in M$ iff $a_{\zeta} \in M$ or $a_{\xi} \in M$.
- The only attackers of $a_{\zeta \downarrow \xi}$ are the arguments $a_{\zeta}$ and $a_{\xi}$. It directly follows that $a_{\zeta \downarrow \xi} \in M$ iff neither $a_{\zeta} \in M$ nor $a_{\xi} \in M$.

These correspondences can be used to show by induction that the newly introduced arguments capture the semantics of the formulas they encode (for all subformulas $\psi$ of $\varphi$ ).

Lemma 27. Let $\varphi$ be a formula over $A$ and $F_{\varphi}$ its associated $A F$. For each stable extension $M$ of $F_{\varphi}$ and $a_{\psi} \in A_{\varphi}$, we have $a_{\psi} \in M$ iff $M \cap A$ is a model of $\psi$.

Proof. Let $M$ be a stable extension of $F$. We use structural induction on $\psi$.
$\psi=\mathrm{T}$ : Trivial: $a_{\top} \in M$ since it has no attackers.
$\psi=\perp$ : Trivial: $a_{\perp} \notin M$ since the set $\left\{a_{\perp}\right\}$ is not conflict-free.
$\psi=p \in A$ : Trivial: $p \in M$ iff $M \models p$ by definition.
$\psi=\neg \xi: a_{\psi} \in M$ iff $a_{\neg \xi} \in M$ iff $a_{\xi} \notin M$ iff $M \not \vDash \xi$ iff $M \models \neg \xi$ iff $M \models \psi$.
$\psi=\zeta \wedge \xi: a_{\psi} \in M$ iff $a_{\zeta \wedge \xi} \in M$ iff $a_{\zeta} \in M$ and $a_{\xi} \in M$ iff $M \models \zeta$ and $M \models \xi$ iff $M \models \zeta \wedge \xi$ iff $M \models \psi$.
$\psi=\zeta \vee \xi: a_{\psi} \in M$ iff $a_{\zeta \vee \xi} \in M$ iff $a_{\zeta} \in M$ or $a_{\xi} \in M$ iff $M \models \zeta$ or $M \models \xi$ iff $M \models \zeta \vee \xi$ iff $M \models \psi$.
$\psi=\zeta \downarrow \xi: a_{\psi} \in M$ iff $a_{\zeta \downarrow \xi} \in M$ iff $a_{\zeta} \notin M$ and $a_{\xi} \notin M$ iff $M \not \vDash \zeta$ and $M \not \vDash \xi$ iff $M \models \zeta \downarrow \xi$ iff $M \models \psi$.

This lets us show the main result of this section, namely that the AF stable extension semantics is universally expressive under projection.

Theorem 28. Let $\varphi$ be a formula over vocabulary $A$ and $F_{\varphi}$ its associated $A F$.

1. For each model $M \subseteq A$ of $\varphi$, there exists a stable extension $E$ of $F_{\varphi}$ with $M=E \cap A$.
2. For each stable extension $E$ of $F_{\varphi}$, the set $E \cap A$ is a model of $\varphi$.

Proof. 1. Let $M \subseteq A$ be a model of $\varphi$. Define the set

$$
E=\left\{a_{\psi} \mid a_{\psi} \in A_{\varphi}, M \models \psi\right\}
$$

Observe that $M=E \cap A$. By presumption, $a_{\varphi} \in E$. It remains to show that $E$ is a stable extension, that is, $E$ is conflict-free and attacks all arguments $b \notin E$.
$E$ is conflict-free: Assume to the contrary that there is an attack $r=(a, b) \in R_{\varphi}$ with $a, b \in E$. By definition, there are only these cases:

- $a$ is arbitrary and $b=\perp$. But then by definition of $E$ we get $M \models \perp$, contradiction.
- $r=\left(p, a_{\neg p}\right)$ or $r=\left(a_{\neg p}, p\right)$ for $p \in A$. But then by definition of $E$ we get $M \models p$ and $M \models \neg p$, contradiction.
- $r=\left(a_{\xi}, a_{\neg \xi}\right)$. But then by definition of $E$ we get $M \models \xi$ and $M \models \neg \xi$, contradiction.
- $r=\left(a_{\neg \zeta}, a_{\zeta \wedge \xi}\right)$ or $r=\left(a_{\neg \xi}, a_{\zeta \wedge \xi}\right)$. Then $M \models \zeta \wedge \xi$, and $M \models \neg \zeta$ or $M \models \neg \xi$, contradiction.
- $r=\left(a_{\zeta \downarrow \xi}, a_{\zeta \vee \xi}\right)$. Then $M \models \zeta \downarrow \xi$, whence $M \models \neg(\zeta \vee \xi)$. But also $M \models \zeta \vee \xi$, contradiction.
- $r=\left(a_{\zeta}, a_{\zeta \downarrow \xi}\right)$ or $r=\left(a_{\xi}, a_{\zeta \downarrow \xi}\right)$. Then $M \models \zeta \downarrow \xi$, and $M \models \zeta$ or $M \models \xi$. But then also $M \models \zeta \vee \xi$, contradiction.
$E$ attacks all arguments not in $E$ : Let $b \in\left(A \cup A_{\varphi} \cup\left\{a_{\perp}\right\}\right) \backslash E$ be an argument. By definition, there is a formula $\psi$ such that $b=a_{\psi}$ and $M \not \vDash \psi$. We use structural induction.
- If $\psi=\perp$ then $a_{\varphi} \in E$ attacks $a_{\perp}$ by definition.
- If $\psi=\neg \xi$, then $M \models \xi$ whence $a_{\xi} \in E$ attacks $a_{\psi}$ by definition.
- If $\psi=\zeta \wedge \xi$, then $M \models \neg \zeta$ or $M \models \neg \xi$ whence $a_{\neg \zeta} \in E$ or $a_{\neg \xi} \in E$. In any case, $E$ attacks $a_{\psi}$ by definition.
- If $\psi=\zeta \vee \xi$, then $M \models \zeta \downarrow \xi$ whence $a_{\zeta \downarrow \xi} \in E$ attacks $a_{\psi}$ by definition.
- If $\psi=\zeta \downarrow \xi$, then $M \models \zeta \vee \xi$ whence $a_{\zeta} \in E$ or $a_{\xi} \in E$.

In any case, $E$ attacks $a_{\psi}$ by definition.
2. Let $E$ be a stable extension of $F_{\varphi}$. Since $E$ is conflict-free, $a_{\perp} \notin E$. Since $E$ is stable, $E$ attacks $a_{\perp}$, which yields $a_{\varphi} \in E$. By Lemma $27, E \cap A$ is a model of $\varphi$.

In particular, $F_{\varphi}$ has no stable extension iff $\varphi$ is unsatisfiable. While this shows that the construction of Definition 8 works as intended, it remains to show that the number of new arguments is at most linear in the formula size. We can even show that the total increase in size is only linear, thus also the number of new arguments is linear.

Proposition 29. For any formula $\varphi$, we find that $\left\|F_{\varphi}\right\| \in O(\|\varphi\|)$.
Proof. We first note that

$$
\begin{aligned}
\left\|F_{\varphi}\right\| & =\left\|\left(A_{\varphi} \cup A_{\perp}, R_{\varphi} \cup\left\{\left(a_{\varphi}, a_{\perp}\right)\right\} \cup R_{\perp}\right)\right\| \\
& =\left|A_{\varphi} \cup A_{\perp}\right|+\left|R_{\varphi} \cup\left\{\left(a_{\varphi}, a_{\perp}\right)\right\} \cup R_{\perp}\right| \\
& =\left|A_{\varphi}\right|+1+\left|R_{\varphi}\right|+2 \\
& =\left|A_{\varphi}\right|+\left|R_{\varphi}\right|+3
\end{aligned}
$$

We now use structural induction on $\varphi$ to show that for all formulas $\varphi$, we find $\left|A_{\varphi}\right| \leq 5 \cdot\|\varphi\|$ and $\left|R_{\varphi}\right| \leq 4 \cdot\|\varphi\|$. It then follows that $\left\|F_{\varphi}\right\| \leq(5+4) \cdot\|\varphi\|+3=9 \cdot\|\varphi\|+3 \in O(\|\varphi\|)$. $\varphi=\mathrm{T}:$

$$
\begin{aligned}
& \left|A_{\top}\right|=\left|\left\{a_{\top}\right\}\right|=1 \leq 5=5 \cdot\|\top\| \\
& \left|R_{\top}\right|=|\emptyset|=0 \leq 4=4 \cdot\|\top\|
\end{aligned}
$$

$\varphi=\perp:$

$$
\begin{aligned}
& \left|A_{\perp}\right|=\left|\left\{a_{\perp}\right\}\right|=1 \leq 5=5 \cdot\|\perp\| \\
& \left|R_{\perp}\right|=\left|\left\{\left(a_{\perp}, a_{\perp}\right)\right\}\right|=1 \leq 4=4 \cdot\|\perp\|
\end{aligned}
$$

$\varphi=a \in A:$

$$
\begin{aligned}
& \left|A_{a}\right|=\left|\left\{a, a_{\neg a}\right\}\right|=2 \leq 5=5 \cdot\|a\| \\
& \left|R_{a}\right|=\left|\left\{\left(a, a_{\neg a}\right),\left(a_{\neg a}, a\right)\right\}\right|=2 \leq 4=4 \cdot\|a\|
\end{aligned}
$$

$\varphi=\neg \xi:$

$$
\begin{aligned}
& \left|A_{\varphi}\right|=\left|A_{\xi} \cup\left\{a_{\neg \xi}\right\}\right| \leq\left|A_{\xi}\right|+1 \leq(5 \cdot\|\xi\|)+1 \leq 5 \cdot(\|\xi\|+1)=5 \cdot\|\varphi\| \\
& \left|R_{\varphi}\right|=\left|R_{\xi} \cup\left\{\left(a_{\xi}, a_{\neg \xi}\right)\right\}\right| \leq\left|R_{\xi}\right|+1 \leq(4 \cdot\|\xi\|)+1 \leq 4 \cdot(\|\xi\|+1)=4 \cdot\|\varphi\|
\end{aligned}
$$

$\varphi=\zeta \wedge \xi:$

$$
\begin{aligned}
\left|A_{\varphi}\right| & \leq\left|A_{\neg \zeta}\right|+\left|A_{\neg \xi}\right|+3 \leq\left(\left|A_{\zeta}\right|+1\right)+\left(\left|A_{\xi}\right|+1\right)+3 \\
& \leq(5 \cdot\|\zeta\|+1)+(5 \cdot\|\xi\|+1)+3=5 \cdot\|\zeta\|+5 \cdot\|\xi\|+5 \\
& =5 \cdot(\|\zeta\|+\|\xi\|+1)=5 \cdot\|\zeta \wedge \xi\|
\end{aligned}
$$

$$
\left|R_{\varphi}\right| \leq\left|R_{\neg \zeta}\right|+\left|R_{\neg \xi}\right|+2 \leq\left(\left|R_{\zeta}\right|+1\right)+\left(\left|R_{\xi}\right|+1\right)+2
$$

$$
\leq(4 \cdot\|\zeta\|+1)+(4 \cdot\|\xi\|+1)+2=4 \cdot\|\zeta\|+4 \cdot\|\xi\|+4
$$

$$
=4 \cdot(\|\zeta\|+\|\xi\|+1)=4 \cdot\|\varphi\|
$$

$\varphi=\zeta \vee \xi:$

$$
\begin{aligned}
\left|A_{\varphi}\right| & \leq\left|A_{\zeta}\right|+\left|A_{\xi}\right|+2 \leq 5 \cdot\|\zeta\|+5 \cdot\|\xi\|+2 \\
& \leq 5 \cdot\|\zeta\|+5 \cdot\|\xi\|+5=5 \cdot(\|\zeta\|+\|\xi\|+1)=5 \cdot\|\varphi\| \\
\left|R_{\varphi}\right| & \leq\left|R_{\zeta}\right|+\left|R_{\xi}\right|+3 \leq(4 \cdot\|\zeta\|)+(4 \cdot\|\xi\|)+3 \\
& \leq 4 \cdot\|\zeta\|+4 \cdot\|\xi\|+4=4 \cdot(\|\zeta\|+\|\xi\|+1)=4 \cdot\|\varphi\|
\end{aligned}
$$

Hence under projection, the AF stable extension semantics can realise as much as propositional logic can. With the results of the previous section ( $\mathrm{AF} \leq_{e} \mathrm{PL}$ ), this means that allowing to introduce a linear number of new vocabulary elements (that are later projected out), all languages considered in this paper are equally (universally) expressive.

However, we must note that equal expressiveness does not mean equal efficiency: When we assume that a knowledge base of size $n$ leads to a search space of size $O\left(2^{n}\right)$, then a linear increase in knowledge base size (that is, from $n$ to $c \cdot n$ for some constant $c$ ) leads to a polynomial increase in search space size (that is, from $O\left(2^{n}\right)$ to $O\left(2^{c \cdot n}\right)=O\left(\left(2^{n}\right)^{c}\right)$.

## 5. Discussion

We compared the expressiveness of abstract argumentation frameworks, abstract dialectical frameworks, normal logic programs and propositional logic. We showed that expressiveness under different semantics varies for the formalisms and obtained a neat expressiveness hierarchy. These results inform us about the capabilities of these languages to encode sets of two-valued interpretations, and help us decide which languages to use for specific applications. Furthermore, we have seen that the results are sensitive to the vocabulary one is permitted to use, as the hierarchy collapses when we allow to introduce even only a linear number of new atoms.

Concerning succinctness, we have shown that ADFs (under model semantics) are exponentially more succinct than normal logic programs (under supported model semantics), and that even bipolar ADFs (under model semantics) - although being less expressive can succinctly express some model sets where equivalent normal logic programs (under supported model semantics) over the same vocabulary must necessarily blow up exponentially in size. It is open whether the converse direction also holds, that is, whether BADFs are exponentially more succinct than logic programs (if $\mathrm{LP}^{s u} \leq_{s} \mathrm{BADF}^{s u}$ ) or the two are just mutually incomparable in terms of succinctness (if $\mathrm{LP}^{s u} \not_{s} \mathrm{BADF}^{s u}$ ). For the stable semantics, relative succinctness of logic programs and BADFs is completely open, partly due to the technical aspect that the two stable semantics are conceptually different, as ADFs in fact employ ultimate stable models (Denecker et al., 2004; Brewka et al., 2013; Strass \& Wallner, 2015). Furthermore, for general ADFs, the computational complexity of the model existence problem of stable semantics is higher than for normal logic programs, ${ }^{10}$ so a succinctness comparison with regard to stable models would be of limited significance.

It is easy to see that AFs have a somewhat special role as they are representationally succinct in any case: for a vocabulary $A_{n}$, there is syntactically no possibility to specify a knowledge base (an AF ) of exponential size, since the largest AF over $A_{n}$ has size $\left\|\left(A_{n}, A_{n} \times A_{n}\right)\right\|=n+n^{2}$ and is thus polynomially large. So anything that can be expressed with an AF can be expressed in reasonable space by definition. However, this "strength" of AFs should be taken with a grain of salt, since they are comparably inexpressive. This can (in addition to the results we presented) already be seen from a simple counting argument: even if all syntactically different AFs over $A_{n}$ were semantically different (which they are not), they could express at most $2^{n^{2}}$ different model sets, which is - for increasing $n$ - negligible in relation to the $2^{2^{n}}$ possible model sets over $A_{n}$.

In their original paper, Gogic et al. (1995) also used a relaxed version of succinctness, where they allowed to introduce a linear number of new variables. It follows from our results in Section 4 that all formalisms we consider here are equally succinct under this relaxed notion.

Parts of the expressiveness results for normal logic programs carry over to further LP classes. For example, canonical logic programs provide a limited form of nesting by allowing literals of the form not not $a$ in rule bodies (Lifschitz et al., 1999). This makes it quite easy to see how normal logic programs under supported semantics can be translated to equivalent canonical logic programs, namely by replacing each positive body atom $a$ by not not $a$ in
10. $\Sigma_{2}^{P}$-hard for ADFs (Strass \& Wallner, 2015) as opposed to in NP for normal LPs (Bidoit \& Froidevaux, 1991; Marek \& Truszczyński, 1991).
all rule bodies. Recently, Shen and Zhao (2014) showed that canonical logic programs and propositional logic programs are succinctly incomparable (under an assumption ${ }^{11}$ ), and also provide interesting avenues for further succinctness studies. We can also add succinctness questions of our own: firstly that of comparing disjunctive logic programs under stable models with general ADFs under stable models, since the two have an equally complex ( $\Sigma_{2}^{P}$-complete) model existence problem (Eiter \& Gottlob, 1995; Brewka et al., 2013). What is more, there have been alternative proposals for stable model semantics for ADFs:

- ours (Strass, 2013) (Definition 3.2, later called "approximate stable models" by Strass \& Wallner, 2015), for which model existence is NP-complete (Strass \& Wallner, 2015) and thus potentially easier than that of the stable models of Brewka et al. (2013) (called "ultimate stable models" by Strass \& Wallner, 2015);
- the "grounded model" semantics by Bogaerts, Vennekens, and Denecker (2015) (Definition 6.8), whose model existence problem is also $\Sigma_{2}^{P}$-complete (Bogaerts et al., 2015);
- the "F-stable model" semantics by Alviano and Faber (2015) (Definition 10).

It follows from Theorem 5.9 of Bogaerts et al. (2015) that grounded models and F-stable models coincide. Still, they are demonstrably different from both approximate and ultimate stable models for ADFs (Alviano \& Faber, 2015), ${ }^{12}$ and their relative succinctness in comparison to normal/disjunctive logic programs is unanalysed.

There is more potential for further work. First of all, a "nice" characterisation of bipolar ADF realisability is still missing; we are unsure whether much improvement over Proposition 11 is possible. Incidentally, for AFs the exact characterisation of compact stable extension realisability constitutes a major open problem (Dunne et al., 2015; Baumann et al., 2014). Second, there are further semantics for abstract dialectical frameworks whose expressiveness could be studied; Dunne et al. (2015) and Dyrkolbotn (2014) already analyse many of them for argumentation frameworks. This work is thus only a start and the same can be done for the remaining semantics. For example the admissible, complete and preferred semantics are all defined for AFs, (B)ADFs and LPs (Strass, 2013; Brewka et al., 2013), and Pührer (2015) has already made a huge step into that direction by characterising realisability. Third, there are further formalisms in abstract argumentation (Brewka et al., 2014) whose expressiveness is by and large unexplored to the best of our knowledge. Finally, the representational succinctness of the subclass of bipolar ADFs (using bipolar propositional formulas to represent them) under supported model semantics is mostly open (cf. Table 2), with some evidence pointing toward meaningful capabilities.

## Acknowledgements

This paper combines, extends and improves results of our previous work (Strass, 2014, 2015b, 2015c). We wish to thank Stefan Woltran for providing a useful pointer to related

[^4]work on realisability in logic programming, Bart Bogaerts for pointing out that grounded models and F-stable models are the same, Jörg Pührer for several suggestions for improvement of the manuscript, and Frank Loebe for helpful discussions. This research was partially supported by Deutsche Forschungsgemeinschaft (DFG, project BR 1817/7-1).

## References

Al-Abdulkarim, L., Atkinson, K., \& Bench-Capon, T. J. M. (2014). Abstract dialectical frameworks for legal reasoning. In Hoekstra, R. (Ed.), Proceedings of the TwentySeventh Annual Conference on Legal Knowledge and Information Systems (JURIX), Vol. 271 of Frontiers in Artificial Intelligence and Applications, pp. 61-70. IOS Press.
Al-Abdulkarim, L., Atkinson, K., \& Bench-Capon, T. J. M. (2015). Evaluating an approach to reasoning with cases using abstract dialectical frameworks. In Proceedings of the Fifteenth International Conference on Artificial Intelligence and Law (ICAIL).

Alviano, M., \& Faber, W. (2015). Stable model semantics of abstract dialectical frameworks revisited: A logic programming perspective. In Yang, Q., \& Wooldridge, M. (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), pp. 2684-2690, Buenos Aires, Argentina. IJCAI/AAAI.

Arora, S., \& Barak, B. (2009). Computational Complexity: A Modern Approach. Cambridge University Press.
Baumann, R., Dvořák, W., Linsbichler, T., Strass, H., \& Woltran, S. (2014). Compact argumentation frameworks. In Proceedings of the Twenty-First European Conference on Artificial Intelligence (ECAI), pp. 69-74, Prague, Czech Republic.

Bidoit, N., \& Froidevaux, C. (1991). Negation by default and unstratifiable logic programs. Theoretical Computer Science, 78(1), 85-112.

Bogaerts, B., Vennekens, J., \& Denecker, M. (2015). Grounded fixpoints and their applications in knowledge representation. Artificial Intelligence, 224, 51-71.
Boppana, R. B. (1986). Threshold functions and bounded depth monotone circuits. Journal of Computer and System Sciences, 32(2), 222-229.
Brewka, G., Dunne, P. E., \& Woltran, S. (2011). Relating the semantics of abstract dialectical frameworks and standard AFs. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI), pp. 780-785. IJCAI/AAAI.

Brewka, G., Ellmauthaler, S., Strass, H., Wallner, J. P., \& Woltran, S. (2013). Abstract dialectical frameworks revisited. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI), pp. 803-809. IJCAI/AAAI.

Brewka, G., \& Gordon, T. F. (2010). Carneades and abstract dialectical frameworks: A reconstruction. In Proceedings of the Third International Conference on Computational Models of Argument (COMMA), Vol. 216 of FAIA, pp. 3-12. IOS Press.
Brewka, G., Polberg, S., \& Woltran, S. (2014). Generalizations of Dung frameworks and their role in formal argumentation. IEEE Intelligent Systems, 29(1), 30-38. Special Issue on Representation and Reasoning.

Brewka, G., \& Woltran, S. (2010). Abstract dialectical frameworks. In Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR), pp. 102-111.
Clark, K. L. (1978). Negation as failure. In Gallaire, H., \& Minker, J. (Eds.), Logic and Data Bases, pp. 293-322. Plenum Press.
Coste-Marquis, S., Konieczny, S., Mailly, J.-G., \& Marquis, P. (2014). On the revision of argumentation systems: Minimal change of arguments statuses. In Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 52-61.
Darwiche, A., \& Marquis, P. (2002). A knowledge compilation map. Journal of Artificial Intelligence Research, 17, 229-264.
Denecker, M., Marek, V. W., \& Truszczyński, M. (2004). Ultimate approximation and its application in nonmonotonic knowledge representation systems. Information and Computation, 192(1), 84-121.
Dimopoulos, Y., Nebel, B., \& Toni, F. (2002). On the computational complexity of assumption-based argumentation for default reasoning. Artificial Intelligence, 141(1/2), 57-78.
Dung, P. M. (1995). On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence, 77, 321-358.
Dunne, P. E., Dvořák, W., Linsbichler, T., \& Woltran, S. (2014). Characteristics of multiple viewpoints in abstract argumentation. In Proceedings of the Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR), pp. 72-81, Vienna, Austria.
Dunne, P. E., Dvořák, W., Linsbichler, T., \& Woltran, S. (2015). Characteristics of multiple viewpoints in abstract argumentation. Artificial Intelligence, 228, 153-178.
Dyrkolbotn, S. K. (2014). How to argue for anything: Enforcing arbitrary sets of labellings using AFs. In Proceedings of the Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR), pp. 626-629, Vienna, Austria.
Eiter, T., Fink, M., Pührer, J., Tompits, H., \& Woltran, S. (2013). Model-based recasting in answer-set programming. Journal of Applied Non-Classical Logics, 23(1-2), 75-104.
Eiter, T., \& Gottlob, G. (1995). On the computational cost of disjunctive logic programming: Propositional case. Annals of Mathematics and Artificial Intelligence, 15(3-4), 289323.

French, T., van der Hoek, W., Iliev, P., \& Kooi, B. (2013). On the succinctness of some modal logics. Artificial Intelligence, 197, 56-85.
Friedman, J. (1986). Constructing $O(n \log n)$ size monotone formulae for the $k$-th elementary symmetric polynomial of $n$ Boolean variables. SIAM Journal on Computing, 15, 641-654.

Gabbay, D. M. (2011). Dung's argumentation is essentially equivalent to classical propositional logic with the Peirce-Quine dagger. Logica Universalis, 5(2), 255-318.

Gaggl, S. A., \& Strass, H. (2014). Decomposing Abstract Dialectical Frameworks. In Parsons, S., Oren, N., \& Reed, C. (Eds.), Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA), Vol. 266 of FAIA, pp. 281-292. IOS Press.

Gaggl, S. A., Rudolph, S., \& Strass, H. (2015). On the computational complexity of naivebased semantics for abstract dialectical frameworks. In Yang, Q., \& Wooldridge, M. (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), pp. 2985-2991, Buenos Aires, Argentina. IJCAI/AAAI.

Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., \& Schneider, M. (2011). Potassco: The Potsdam Answer Set Solving Collection. AI Communications, $24(2), 105-124$. Available at http://potassco.sourceforge.net.
Gelfond, M., \& Lifschitz, V. (1988). The stable model semantics for logic programming. In Proceedings of the International Conference on Logic Programming (ICLP), pp. 1070-1080. The MIT Press.

Gogic, G., Kautz, H., Papadimitriou, C., \& Selman, B. (1995). The comparative linguistics of knowledge representation. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI), pp. 862-869. Morgan Kaufmann.
Jukna, S. (2012). Boolean Function Complexity: Advances and Frontiers, Vol. 27 of Algorithms and Combinatorics. Springer.
Lifschitz, V., \& Razborov, A. (2006). Why are there so many loop formulas?. ACM Transactions on Computational Logic, 7(2), 261-268.

Lifschitz, V., Tang, L. R., \& Turner, H. (1999). Nested expressions in logic programs. Annals of Mathematics and Artificial Intelligence, 25(3-4), 369-389.

Lin, F., \& Zhao, Y. (2004). ASSAT: Computing answer sets of a logic program by SAT solvers. Artificial Intelligence, 157(1-2), 115-137.

Linsbichler, T. (2014). Splitting abstract dialectical frameworks. In Parsons, S., Oren, N., \& Reed, C. (Eds.), Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA), Vol. 266 of FAIA, pp. 357-368. IOS Press.
Marek, V. W., \& Truszczyński, M. (1991). Autoepistemic logic. Journal of the ACM, 38(3), 587-618.

Osorio, M., Zepeda, C., Nieves, J. C., \& Cortés, U. (2005). Inferring acceptable arguments with answer set programming. In Proceedings of the Sixth Mexican International Conference on Computer Science (ENC), pp. 198-205.

Polberg, S. (2014). Extension-based semantics of abstract dialectical frameworks. In Endriss, U., \& Leite, J. (Eds.), Proceedings of the Seventh European Starting AI Researcher Symposium (STAIRS), Vol. 264 of FAIA, pp. 240-249. IOS Press.
Polberg, S., Wallner, J. P., \& Woltran, S. (2013). Admissibility in the abstract dialectical framework. In Leite, J., Son, T. C., Torroni, P., van der Torre, L., \& Woltran, S. (Eds.), Proceedings of the Fourteenth International Workshop on Computational Logic in Multi-Agent Systems (CLIMA XIV), Vol. 8143 of LNAI, pp. 102-118. Springer.

Pührer, J. (2015). Realizability of three-valued semantics for abstract dialectical frameworks. In Yang, Q., \& Wooldridge, M. (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), pp. 3171-3177. IJCAI/AAAI, Buenos Aires, Argentina.

Shen, Y., \& Zhao, X. (2014). Canonical logic programs are succinctly incomparable with propositional formulas. In Proceedings of the Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR), pp. 665-668, Vienna, Austria.

Strass, H. (2013). Approximating operators and semantics for abstract dialectical frameworks. Artificial Intelligence, 205, 39-70.

Strass, H. (2014). On the relative expressiveness of argumentation frameworks, normal logic programs and abstract dialectical frameworks. In Konieczny, S., \& Tompits, H. (Eds.), Proceedings of the Fifteenth International Workshop on Non-Monotonic Reasoning (NMR).
Strass, H. (2015a). Instantiating rule-based defeasible theories in abstract dialectical frameworks and beyond. Journal of Logic and Computation, Advance Access published 11 February 2015, http://dx.doi.org/10.1093/logcom/exv004.
Strass, H. (2015b). The relative expressiveness of abstract argumentation and logic programming. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI), pp. 1625-1631, Austin, TX, USA.
Strass, H. (2015c). Representational succinctness of abstract dialectical frameworks. In Black, E., Modgil, S., \& Oren, N. (Eds.), Proceedings of the Third International Workshop on Theory and Applications of Formal Argumentation (TAFA).

Strass, H., \& Wallner, J. P. (2015). Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. Artificial Intelligence, 226, 34-74.

Tseitin, G. S. (1968). On the complexity of derivations in the propositional calculus. Structures in Constructive Mathematics and Mathematical Logic, Part II, Seminars in Mathematics (translated from Russian), 115-125.


[^0]:    1. However, the stable model semantics becomes universally expressive once we allow nested expressions of the form "not not $p$ " in rule bodies (Lifschitz, Tang, \& Turner, 1999; Lifschitz \& Razborov, 2006).
[^1]:    4. This operator is closely related to the ultimate approximation operators of Denecker, Marek, and Truszczyński (2004), as we observed earlier (Strass, 2013).
[^2]:    5. Already for complexity reasons, we cannot expect that this translation is also faithful for the stable semantics. And indeed, the ADF $D=\left(\{a\},\{(a, a)\},\left\{\varphi_{a}=a \vee \neg a\right\}\right)$ has a stable model $\{a\}$ while its standard logic program $P(D)=\{a \leftarrow\{a\}, a \leftarrow\{$ not $a\}\}$ has no stable model. However, it holds that $s t(P(D)) \subseteq s t(D)$ (Denecker et al., 2004; Strass, 2013).
[^3]:    9. We remark that the three open problems in Table 2 are really only two: It is easy to show that ADFs and propositional logic behave equivalently in relation to bipolar ADFs, since they are equally expressive and equally succinct; that is, it holds that $\mathrm{ADF}^{s u} \leq_{s} \mathrm{BADF}^{s u}$ if and only if $\mathrm{PL} \leq_{s} \mathrm{BADF}^{s u}$.
[^4]:    11. $\mathrm{P} \nsubseteq \mathrm{NC}_{\text {/poly }}^{1}$, the Boolean circuit equivalent of the assumption NP $\nsubseteq \mathrm{P}$.
    12. In the terminology of Alviano and Faber (2015), approximate stable models (Strass, 2013) are called S-stable models and ultimate stable models (Brewka et al., 2013) are called B-stable models. Both are shown to be different from F-stable models.
