

# Abstract Dialectical Frameworks are Boolean Networks<sup>\*</sup>

Jesse Heyninck<sup>1,2</sup>, Matthias Knorr<sup>3</sup> and João Leite<sup>3</sup>

<sup>1</sup>Open Universiteit, the Netherlands

<sup>2</sup>University of Cape Town, South-Africa

<sup>3</sup>NOVA LINCS, NOVA University Lisbon

## Abstract

Abstract dialectical frameworks are a unifying model of formal argumentation, where argumentative relations between arguments are represented by assigning acceptance conditions to atomic arguments. Their generality allows them to cover a number of different approaches with varying forms of representing the argumentation structure. Boolean regulatory networks are used to model the dynamics of complex biological processes, taking into account the interactions of biological compounds, such as proteins or genes. These models have proven highly useful for comprehending such biological processes, allowing to reproduce known behaviour and testing new hypotheses and predictions *in silico*, for example in the context of new medical treatments. While both these approaches stem from entirely different communities, it turns out that there are striking similarities in their appearance. In this paper, we study the relation between these two formalisms revealing their commonalities as well as their differences, and introducing a correspondence that allows to establish novel results for the individual formalisms.

## 1. Introduction

Formal argumentation is one of the major approaches to knowledge representation and reasoning. In the seminal paper [1], Dung introduced *abstract argumentation frameworks*, conceived as directed graphs where nodes represent arguments and edges between nodes represent attacks. Their meaning is given by so-called *argumentation semantics* that determine which sets of arguments can be reasonably upheld together given such an argumentation graph. Various authors have since remarked that other relations between arguments are worth consideration, such as, for example, a dual *support* relation as in *bipolar argumentation frameworks* [2]. The last decades witnessed a proliferation of extensions of the original formalism that has often made it hard to compare the dialects of the different resulting argumentation formalisms. In an attempt to cope with the resulting number of dialects and unify them, *abstract dialectical frameworks* (in short, ADFs) were introduced [3]. Just like abstract argumentation frameworks, ADFs are also directed graphs. However, in ADFs, edges between nodes do not necessarily represent attacks, but can encode *any* relationship between arguments. Such generality is achieved by associating an *acceptance condition* with each argument, represented as a Boolean formula over the parents of the argument, expressing the conditions under which the argument can be accepted. ADFs offer a general framework for argumentation-based inference as they are able to capture all of the major semantics of abstract argumentation [4], and even normal logic programs [3]. The following example illustrates a simple ADF.

**Example 1.** You are considering your travel plans for the upcoming conference summer. If you manage to write a paper, it will be suitable for a conference in *texas* or in *vietnam*. You are only allowed to submit the paper to the conference in Vietnam if you do not submit it to another conference. (but the conference in Texas does allow for submission of papers submitted elsewhere). Both conferences will require you to apply for travelling funds. This example can be expressed formally in the ADF given in Fig. 1.a). Interactions

22nd International Workshop on Nonmonotonic Reasoning, November 2-4, 2024, Hanoi, Vietnam

<sup>\*</sup>This paper has been accepted at LPNMR 2024.

✉ jesse.heyninck@ou.nl (J. Heyninck); mkn@fct.unl.pt (M. Knorr);

jleite@fct.unl.pt (J. Leite)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

between atoms are expressed by arrows (e.g. writing a paper influences attendance of a conference in both Vietnam and Texas), and the acceptance conditions express these interactions more precisely. E.g.,  $C_v = \neg t \wedge p$  expresses conference attendance in Vietnam is possible if one has a paper and did not submit it to the conference in Texas.

In systems biology, *Biological regulatory networks* encode interactions between biological specimens or compounds, such as proteins or genes, and their interactions, to acquire a better understanding of the complex processes that take place in cells, as doing so may lead to discoveries and new theories about living organisms. To abstract from actual concentration values, and use thresholds to represent whether a compound is active or inactive, logical models [5, 6] are often used instead of quantitative models. Because of that, they require far less information than quantitative models and are therefore more adequate to deal with incomplete, imprecise, and noisy information regarding the biological system. Among these, Boolean (logical) models or networks (BNs), have been extensively used to reproduce known behaviour and test new hypotheses *in silico*, e.g., as models of gene regulation networks and other biological systems [7].

**Example 2.** Consider the following toy scenario of a biosphere where four species are potentially present: native quagga mussels, invasive zebra mussels, algae, and fish. These species interact as follows: fish feed on mussels, zebra-mussels outcompete quagga, and both kinds of mussels feed on algae. These interactions can be represented using the biological network in Fig. 1.b). Notice that this is a simplified representation of a biological network, meant merely to ease understanding. Influences are represented by arrows, marked with a “+” if the influence is positive (e.g. since fish feed on mussels), and a “-” if the influence is negative (e.g. since zebra mussels outcompete quagga mussels). Furthermore, the precise nature of these interactions is encoded by Boolean functions on the right-hand side of the figure. For example,  $f_q = \neg z \wedge a$  expresses that quagga mussels will be alive if there are no zebra-mussels and there are algae.

These examples show striking similarities between these models of two very different subject matters: ADFs and BNs. Syntactically, they both use directed graphs to encode interactions, and Boolean formulae to express the precise nature of such interactions. The open question is whether such similarities extend from the syntactic to the semantic level.

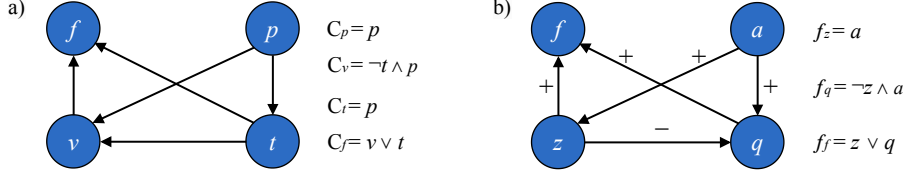


Figure 1: Examples of a) an ADF for Example 1, and b) a BN for Example 2.

If that were the case, it would open up a fruitful ground for the cross-fertilization of both fields, both theoretically, where established results from one could provide new insights for the other, but also practically, e.g., by allowing implementations for one to be used by the other.

In this paper, we compare ADFs and BNs. After formally introducing ADFs in Sec. 2, in Sec. 3 we introduce BNs and show that the similarities with ADFs extend from the syntactic to the semantic level, by pointing in a formally precise way what are the equivalent notions, but also what the differences are. Our results show that ADFs are essentially BNs, creating a fruitful ground for investigating synergies, as illustrated in Sec. 4. We then conclude in Sec. 5.<sup>1</sup>

## 2. Abstract Dialectical Frameworks

We recall ADFs following loosely the notation from [3]. An ADF  $D$  is a tuple  $D = (\text{At}, L, C)$  where  $\text{At}$  is a finite set of atoms, representing arguments or statements,  $L \subseteq \text{At} \times \text{At}$  is a set of links, representing dependencies or attacks from one argument against another, and  $C = \{C_s\}_{s \in \text{At}}$  is a set of total functions (also called acceptance functions)  $C_s : 2^{\text{par}_D(s)} \rightarrow \{1, 0\}$  for each  $s \in \text{At}$  with  $\text{par}_D(s) = \{s' \in \text{At} \mid (s', s) \in L\}$  and truth values true (1) and false (0). An acceptance function  $C_s$  defines the cases when the statement  $s$  can be accepted (is true), depending on the acceptance status of its parents in  $D$ . We often identify an acceptance function  $C_s$  by its equivalent *acceptance condition* which models the acceptable cases as a propositional formula over  $\text{At}$  and the usual Boolean connectives  $\wedge$  (and),  $\vee$  (or),  $\neg$  (negation) and  $\rightarrow$  (material implication). Also, the set of links of  $D$  is completely determined by  $C_s$  and sometimes left implicit.

**Example 3.** In Ex. 1, we find an ADF  $D = (\{f, p, t, v\}, L, C)$  with  $L$  and  $C$  as in Fig 1.a). Here, we omit reflexive arrows (e.g.  $(p, p)$ ) to avoid clutter. An acceptance condition like  $C_v = \neg t \wedge p$  can be read as “ $v$  is accepted if  $t$  is not accepted and  $p$  is accepted”.

Interpretations can be used to formally assign meaning to these acceptance conditions. An *interpretation* (also called *possible world*)  $\omega$  is a function  $\omega : \text{At} \rightarrow \{1, 0\}$ . Let  $\mathcal{V}^2(\text{At})$  denote the set of all interpretations for  $\text{At}$ . We simply write  $\mathcal{V}^2$  if the set of atoms is implicitly given. An interpretation  $\omega$  *satisfies* (or is a *model* of) an atom  $a \in \text{At}$ , denoted by  $\omega \models a$ , if and only if  $\omega(a) = 1$ . The satisfaction relation  $\models$  is extended to formulas as usual. Then, an interpretation  $\omega$  is a *two-valued model* of an ADF  $D$  if, for all  $s \in \text{At}$ ,  $\omega \models s$  iff  $\omega \models C_s$ . For sets of formulas  $\Phi$ , we also define  $\omega \models \Phi$  if and only if  $\omega \models \phi$  for every  $\phi \in \Phi$ , and the set of models  $\text{Mod}^2(\Phi) = \{\omega \in \mathcal{V}^2(\text{At}) \mid \omega \models \Phi\}$  for every set of

formulas  $\Phi$ . A set of formulas  $\Phi_1$  *entails* another set of formulas  $\Phi_2$ , denoted by  $\Phi_1 \vdash \Phi_2$ , if  $\text{Mod}^2(\Phi_1) \subseteq \text{Mod}^2(\Phi_2)$ . A formula  $\phi$  is a tautology if  $\text{Mod}^2(\phi) = \mathcal{V}^2(\text{At})$  and inconsistent if  $\text{Mod}^2(\phi) = \emptyset$ . We compare two possible worlds  $\omega_1$  and  $\omega_2$  by:  $\omega_1 \leq \omega_2$  iff for every  $\alpha \in \text{At}$ ,  $\omega_1(\alpha) = 1$  implies  $\omega_2(\alpha) = 1$ .

Commonly though, an ADF  $D = (\text{At}, L, C)$  is interpreted through 3-valued interpretations  $\nu : \text{At} \rightarrow \{1, 0, \text{U}\}$  adding truth value undecided (U). We denote the set of all 3-valued interpretations over  $\text{At}$  by  $\mathcal{V}^3(\text{At})$ . We define the information order  $\leq_i$  over  $\{1, 0, \text{U}\}$  by making U the minimal element:  $\text{U} <_i 1$  and  $\text{U} <_i 0$ , and  $\dagger \leq_i \ddagger$  iff  $\dagger <_i \ddagger$  or  $\dagger = \ddagger$  for any  $\dagger, \ddagger \in \{1, 0, \text{U}\}$ . This order is lifted point-wise as follows (given  $\nu, \nu' \in \mathcal{V}^3(\text{At})$ ):  $\nu \leq_i \nu'$  iff  $\nu(s) \leq_i \nu'(s)$  for every  $s \in \text{At}$ . The set of two-valued interpretations extending a 3-valued interpretation  $\nu$  is defined as  $[\nu]^2 = \{\omega \in \mathcal{V}^2(\text{At}) \mid \nu \leq_i \omega\}$ . Given a set of 3-valued interpretations  $V \subseteq \mathcal{V}^3(\text{At})$ ,  $\Pi_i V$  is the 3-valued interpretation defined via  $\Pi_i V(s) = \dagger$  if for every  $\nu \in V$ ,  $\nu(s) = \dagger$ , for any  $\dagger \in \{1, 0, \text{U}\}$ , and  $\Pi_i V(s) = \text{U}$  otherwise.

All major semantics of ADFs single out three-valued interpretations in which the truth value of every atom  $s \in \text{At}$  is, in some sense, in alignment or agreement with the truth value of the corresponding condition  $C_s$ . The  $\Gamma$ -function enforces this intuition by mapping an interpretation  $\nu$  to a new interpretation  $\Gamma_D(\nu)$ , which assigns to every atom  $s$  exactly the truth value assigned by  $\nu$  to  $C_s$ , i.e.:

$$\Gamma_D(\nu) : \text{At} \rightarrow \{1, 0, \text{U}\} \text{ where } s \rightarrow \Pi_i \{\omega(C_s) \mid \omega \in [\nu]^2\}.$$

We also need to define the *reduct*  $D^\nu$  of  $D$  given  $\nu$ , i.e.,  $D^\nu = (\text{At}^\nu, L^\nu, C^\nu)$  with: (i)  $\text{At}^\nu = \{s \in \text{At} \mid \nu(s) = 1\}$ , (ii)  $L^\nu = L \cap (\text{At}^\nu \times \text{At}^\nu)$ , and (iii)  $C^\nu = \{C_s[\phi/\psi] \mid \nu(\phi) = 0\}/0 \mid s \in \text{At}^\nu\}$ , where  $C_s[\phi/\psi]$  is obtained by substituting every occurrence of  $\phi$  in  $C_s$  by  $\psi$ .

**Definition 1.** Let  $D$  be an ADF with  $\nu \in \mathcal{V}(\text{At})$  a 3-valued interpretation. Then,  $\nu$  is *admissible* for  $D$  iff  $\nu \leq_i \Gamma_D(\nu)$ ;  $\nu$  is *complete* for  $D$  iff  $\nu = \Gamma_D(\nu)$ ;  $\nu$  is *preferred* for  $D$  iff  $\nu$  is  $\leq_i$ -maximal among all admissible models;  $\nu$  is *grounded* for  $D$  iff  $\nu$  is  $\leq_i$ -minimal among all complete models; and  $\nu$  is *stable* iff  $\nu$  is a two-valued model of  $D$  and  $\{s \in \text{At} \mid \nu(s) = 1\} = \{s \in \text{At} \mid w(s) = 1\}$  where  $w$  is the grounded model of  $D^\nu$ . We denote by  $2V(D)$ ,  $\text{adm}(D)$ ,  $\text{cmp}(D)$ ,  $\text{prf}(D)$ ,  $\text{grnd}(D)$ , and  $\text{stb}(D)$  the sets of two-valued, admissible, complete, preferred, grounded, and stable models of  $D$ .

It was been shown that  $\text{stb}(D) \subseteq 2V(D) \subseteq \text{prf}(D) \subseteq \text{cmp}(D) \subseteq \text{adm}(D)$  as well as  $\text{grnd}(D) \subseteq \text{cmp}(D)$ .

**Example 4** (Ex. 1 ctd.). The ADF in Ex. 1 has three complete models  $\nu_1, \nu_2, \nu_3$  with:  $\nu_1(p) = 1, \nu_1(v) = 0, \nu_1(t) = 1$  and  $\nu_1(f) = 1; \nu_2(s) = 0$  and  $\nu_3(s) = \text{U}$  for all  $s \in \text{At}$ . An admissible model that is not complete is  $\nu_4$  with  $\nu_4(p) = 1, \nu_4(v) = \nu_4(t) = \nu_4(f) = \text{U}$ .  $\nu_3$  is the grounded model, whereas  $\nu_1$  and  $\nu_2$  are both preferred and two-valued, and only  $\nu_2$  is stable.

<sup>1</sup>An extended version of this paper with all the proofs is available at <https://arxiv.org/abs/2407.02055>

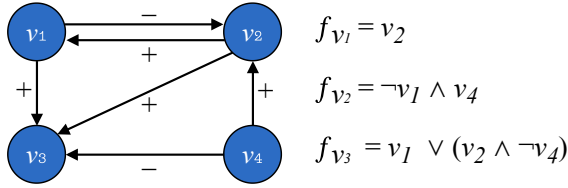


Figure 2: Example of a Boolean logical model.

### 3. Boolean Networks

In this section, we recall Boolean networks as known from the literature (see e.g., [8]), first looking at the syntax and then focussing on their semantics. During this presentation, we will establish sometimes surprising connections to ADFs as well as notable differences between these two formalisms.

#### 3.1. Syntax

Boolean networks utilize a regulatory graph to represent the compounds in the biological process and the principal interactions between them.

**Definition 2.** A *regulatory graph* is a directed graph  $G = (V, E)$ , where  $V = \{v_1, \dots, v_n\}$  is the set of vertices (nodes) representing the regulatory compounds, and  $E = \{(u, v, s) : u, v \in V, s \in \{+, -, \pm\}\}$  is the set of signed edges representing the *interactions* between compounds.

An edge with  $s = +$  is called *positive interaction* (or *activation*), representing that  $u$  activates  $v$ , while an edge with  $s = -$  is called *negative interaction* (or *inhibition*), representing that  $u$  inhibits  $v$ . Occasionally,  $u$  may (in combination with different compounds) both activate and inhibit another compound  $v$ , which is represented with an edge with  $s = \pm$ . A node with no incoming edges is called *input node*, representing external stimuli, whose values do not change.

**Example 5.** Fig. 2 shows regulatory graph  $G = (V, E)$  with  $V = \{v_1, v_2, v_3, v_4\}$  and  $E = \{(v_1, v_2, -), (v_1, v_3, +), (v_2, v_1, +), (v_2, v_3, +), (v_4, v_2, +), (v_4, v_3, -)\}$ .

Boolean logical models then add regulatory functions for each compound to specify how different compounds that affect the same node interact with each other for that node's activation.

**Definition 3.** A *Boolean logical model*  $M$  of a regulatory network is defined as a tuple  $(V, F)$  where  $V = \{v_1, v_2, \dots, v_n\}$  is the set of variables representing the regulatory compounds of the network such that  $v_i$  can be assigned to a value in  $\{0, 1\}$ , and  $F = \{f_1, f_2, \dots, f_n\}$  is the set of Boolean functions such that  $f_i$  defines the value of  $v_i$  and where  $f_i = v_i$  if  $v_i$  is an input node.

Regulatory functions of input nodes may sometimes be omitted (cf. Fig. 2), which means that  $f_i = v_i$ , but commonly, we use the explicit representation.

**Example 6.** Fig. 2 presents a Boolean logical model with  $G$  from Ex. 5 and regulatory functions for  $G$  on the right.

We can observe that, syntactically, a Boolean network is strikingly similar to an abstract dialectical framework. The only difference is that, unlike the edges in the regulatory

graph of a BN, links in ADFs commonly do not mention explicitly whether an argument is attacking or supporting. Still, this can be extracted from the corresponding acceptance conditions, and in some literature [20], the so-called *polarity* of links is explicitly represented (see also Section 3.3). Similarly, the exact representation of regulatory graphs differs in the literature. Using our notation here, we establish this connection between Boolean networks and ADFs formally.

**Definition 4.** Let  $M = (V, F)$  be a Boolean logical model of a regulatory network with regulatory graph  $G = (V, E)$ . We define the corresponding ADF  $D_{M,G} = (V, L, F)$  with  $L = \{(u, v) \mid (u, v, s) \in E\}$ .

Let  $D = (\text{At}, L, C)$  be an ADF. We define the corresponding Boolean logical model  $M_D$  of a regulatory network as  $M_D = (\text{At}, C)$ . The corresponding regulatory graph  $G_D = (\text{At}, E)$  can be obtained from  $C$  in NNF with  $E = \{(u, v, +) \mid u \in C_v, \neg u \notin C_v\} \cup \{(u, v, -) \mid \neg u \in C_v, u \notin C_v\} \cup \{(u, v, \pm) \mid u \in C_v, \neg u \in C_v\}$ .

The requirement for the acceptance conditions to be in Negation Normal Form (NNF) is necessary to include the correct edges in  $G_D$ . Alternatively, one can determine the polarity using Definition 11 below. Either way, as we will see next, the semantics of Boolean networks is uniquely determined by the compounds and their Boolean functions, i.e., the Boolean logical model, and the regulatory graph can be left implicit.

#### 3.2. Dynamics

BNs allow us to capture the changes over time in a biological process based on the interactions of the various compounds involved, which should correctly represent the dynamics observable in the real system. We will see that the study of dynamics in BNs correspond to *semantics* of ADFs. We start with network states that are used to represent the (current) activations of a network's compounds.

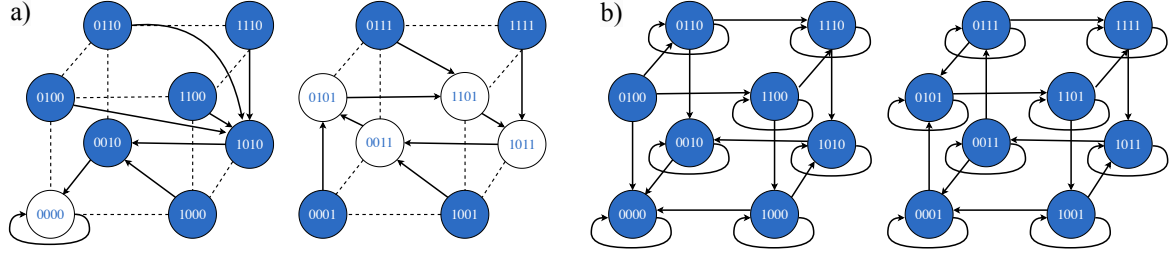
**Definition 5.** The *network state* of a BN with  $n$  compounds is a vector  $S = (v_1, \dots, v_n)$  where  $v_i$  is the value of the variable representing the  $i$ -th compound of the network.

Clearly, for Boolean logical models, the number of different states in a network is given by  $2^n$ . E.g., if nodes 1 and 3 in Ex. 6 are active and the other two are not, then the state will be represented by 1010. Similar representations are used for interpretations of ADFs (by ordering the atoms), and it is clear that the states in BNs correspond to possible world of ADFs.

The update of the  $i$ -th compound  $v_i$  of a network from one discrete time point  $t$  to the next is then defined as  $v_i(t+1) = f_i(S(t))$  for state  $S(t)$  at time  $t$ , which clearly exactly corresponds to the evaluation of an acceptance condition in ADFs. We can then use state transition graphs [9] to describe how networks, and thus the modelled biological systems, evolve over time.

**Definition 6.** A State Transition Graph (STG) is a directed graph  $G_{STG} = (S, T)$  where  $S$  is the set of vertices representing the different states of the network, and  $T$  is the set of edges representing the viable transitions between states according to given update scheme.

Two update schemes are employed to update the values of nodes in a BN: the synchronous and the asynchronous



**Figure 3:** STG of the model in Ex. 6 for a) synchronous updates and b) asynchronous updates, both with  $v_4$  inactive on the left, and active on the right (adapted from [12]).

updating scheme [10, 11]. Note that, given a Boolean logical model, the state transition graphs for the synchronous updating scheme and asynchronous updating scheme are uniquely determined.

In the synchronous updating scheme, at each time step, all compounds are updated simultaneously, i.e., given a state  $S = (v_1, \dots, v_n)$ , the new state is obtained as  $S' = (f_1(S), \dots, f_n(S))$ . Each network state has at most one successor (cf. Fig. 3.a), which is sometimes argued to be biologically less realistic and less accurate for analysis. Yet, synchronous updates are still regularly used [13], and many of the concepts below, such as trap spaces, are independent of the used scheme.

In the asynchronous updating scheme, at each time step, one or more regulatory functions may be applied [13]. This is closer to what is observable in real systems, since these changes seldomly tend to take place simultaneously. With  $n$  compounds in a network, and the frequently used, particular case of exactly one function being applied at each time step, each state can have at most  $n$  possible state transitions (including a transition to itself - cf. Fig. 3.b).

There are certain cycles of states in which these networks reside most of the time. These cycles are a trap of sorts, since as soon as a network enters a cycle's state, it is unable to exit the cycle. These traps, called *attractors*, are linked to many important cellular processes, such as phenotypes, cell cycle phases, cell growth, differentiation, apoptosis, and more [14]. Because the number of states in a network is finite, transitions from all states eventually lead to an attractor. All states that lead to a certain attractor form its attraction basin.

Attractors have other relevant characteristics. One such characteristic is that, from any state belonging to an attractor, it is possible to find a path of attractor states to any other state in that attractor. Another important property is that there are no state transitions from attractor states to states outside of the attractor. Attractors can also be classified under different types. The left image of Fig. 3.a) shows an example of a *stable state* (or *point attractor*) in the white-colored  $0000$  state. Stable states are attractors that contain only a single state. When that is not the case, the attractor is denoted as a *cyclic* (or *complex*) *attractor*, visualized in the right image of Fig. 3.a). There, we have a complex attractor comprised of four states:  $0101$ ,  $1101$ ,  $1011$  and  $0011$  [15].

**Definition 7.** Given an STG  $G_{STG} = (S, T)$ , a set  $S' \subseteq S$  is a *trap set* of  $G_{STG}$  if, for every  $\omega \in S'$ ,  $(\omega, \omega') \in T$  implies  $\omega' \in S'$ . An *attractor* is a  $\subseteq$ -minimal trap set. A *stable state* of  $G$  is a singleton trap set of  $G$ .

We obtain that stable states of a transition graph  $G$  under synchronous update scheme coincide with the two-valued

models of the corresponding ADF.

**Proposition 1.** Let  $G_{STG}$  be the synchronous state transition graph of the Boolean Model  $M$  with regulatory graph  $G$ , and  $D_{M,G}$  the corresponding ADF. Then,  $\omega$  is a stable state of  $G_{STG}$  iff  $\omega$  is a two-valued model of  $D_{M,G}$ .

To the best of our knowledge, the more general notion of a trap set has not been investigated in the context of ADFs. This is not surprising, as it has a clear meaning and use in biological networks, but not so much in argumentation:

**Example 7.** Consider the ADF  $D = (\{a, b, c\}, L, C)$  with  $C_a = \neg c$ ,  $C_b = \neg a$  and  $C_c = \neg b$ . Then  $\{000, 111\}$  is a trap set and an attractor, but their argumentative interpretation is not clear: if we interpret  $a$ ,  $b$ , and  $c$  as arguments that attack each other, the stability under transitions of  $\{000, 111\}$ , interesting in a biological interpretation, is of less interest in argumentation.

More surprisingly, we will now see that many other semantics for ADFs have a natural counterpart in Boolean networks. For example, there has been a lot of interest in so-called *subspaces* of regulatory graphs [8], which are sets of interpretations for which assignments of some variables are fixed. These are represented as assignments  $m : V \mapsto \{0, 1, \star\}$  of the variables  $V$  to the “classical” values 0 and 1 or the value  $\star$ , intuitively representing that the respective variable has no fixed assignment. This means that for every  $v \in V$  for which  $m(v) = \star$ , the subspace contains both states that assign 0 to  $v$  and states that assign 1 to  $v$ . Trap spaces have received a lot of attention in the literature on Boolean networks as finding them is computationally easier than finding *any* trap set [16], while they are still guaranteed to contain a trap set [8].

**Definition 8.** A subspace  $m$  of a regulatory graph  $(V, E)$  is a mapping  $m : V \mapsto \{0, 1, \star\}$ . We call  $v \in V$  *fixed* if  $m(v) \in \{0, 1\}$  and *free* if  $m(v) = \star$ , and  $D_m$  is the set of all fixed variables in  $m$ . We let  $S[m] := \{s \in S \mid \forall v \in D_m : s(v) = m(v)\}$ .

We sometimes abuse notation and identify a subspace  $m$  with its expanded variant  $S[m]$ .

An interesting observation is that subspaces, interpreted as their representative set of Boolean valuations  $S[m]$ , correspond exactly to the completions of the subspaces, interpreted as three-valued interpretations:

**Proposition 2.** Let  $m$  be a subspace  $m$ , and  $\nu_m : V \mapsto \{0, 1, \mathbf{U}\}$  defined by

$$\nu_m(v) = \begin{cases} m(v) & \text{if } v \in D_m \\ \mathbf{U} & \text{otherwise.} \end{cases} \quad (1)$$

It holds that  $[\nu_m]^2 = S[m]$ .

Subspaces that are also trap sets are of special interest in the literature on Boolean networks.

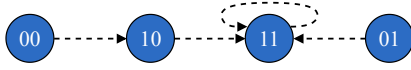
**Definition 9.** A *trap space* is a subspace  $m$  for which  $S[m]$  is a trap set.

Trap spaces, interpreted as three-valued interpretations (see Proposition 2) correspond to admissible interpretations:

**Proposition 3.** Let  $M$  be a Boolean Model with regulatory graph  $G$ , and  $D_{M,G}$  the corresponding ADF:  $m$  is a trap space of  $M$  iff  $\nu_m$  is admissible in  $D_{M,G}$ .

On the other hand, the complete semantics does not seem to have a direct counterpart in Boolean networks.

**Example 8.** Consider the ADF  $D = (\{a, b\}, L, C)$  with  $C_a = \{a \vee \neg a\}$  and  $C_b = b$ . Let us take a look at the STG for synchronous updates first:



There are three trap spaces:  $\star\star$ ,  $\star 1$ , and  $11$ . By Prop. 3, this corresponds to the interpretations  $UU$  and  $11$ . However, the unique complete interpretation is  $11$ .

We might conjecture that the complete interpretations correspond to the minimal trap spaces, but that is not the case either.

**Example 9.** Consider the ADF  $D = (\{a\}, L, C)$  with  $C_a = \{a\}$ . There are three trap spaces:  $\star$ ,  $1$ , and  $0$ . The corresponding interpretations  $U$ ,  $1$  and  $0$  are not only admissible but also complete. However,  $U$  is not a minimal trap space.

We further note that there has been some interest in maximal and minimal trap spaces [15] in the literature on BNs. In more detail, trap sets are compared as follows:  $m_1 \leq m_2$  iff  $S[m_1] \subseteq S[m_2]$ . It can be easily observed that this is the reverse of the information order  $\leq_i$  known from ADFs:

**Proposition 4.** Let  $m_1, m_2$  be two subspaces. Then  $m_1 \leq m_2$  iff  $\nu_{m_2} \leq_i \nu_{m_1}$ .

*Minimal* trap spaces are trap spaces  $m_1$  such that there is no trap space  $m_2 < m_1$ . *Maximal* trap spaces are trap spaces  $m_1$  s.t. there is no trap space  $m_2 > m_1$  and  $S[m_1] \neq S$ , i.e., the trivial trap space is excluded by definition.

**Proposition 5.** Let  $M$  be a BN with regulatory graph  $G$ , and  $D_{M,G}$  the corresponding ADF:  $m$  is a minimal trap space of  $M$  iff  $\nu_m$  is preferred in  $D_{M,G}$ .

However, maximal trap spaces do not correspond to the grounded model. The reason is that in BNs, the trivial trap space is excluded. E.g., in Ex. 3,  $\nu_3$  is the grounded model, but does not correspond to a maximal trap space as it is trivial trap space. Finally, we note that the concept of stable model from ADFs has no clear counterpart in BNs. Indeed, the main motivation behind stable models is to exclude self-supporting arguments (see e.g., Ex. 4 where  $\nu_1$  with  $p$  supporting itself via  $C_p = p$  is excluded). In BNs, there is nothing *a priori* wrong with such self-supporting stable states, and it might often even have a clear biological meaning, e.g., since algae are self-reproducing.

### 3.3. Subclasses of Boolean Networks

An interesting observation is that both the literature on ADFs and the literature on Boolean Networks has identified certain subclasses of frameworks for which the computational complexity of computational tasks decreases. In fact, both strands of literature have identified the same subclass! In the literature on ADFs, these frameworks are called *bipolar* ADFs, whereas in the literature on Boolean networks, they are called *sign-definite*.

**Definition 10.** A Boolean function  $f : \{0, 1\}^n \mapsto \{0, 1\}$  is *increasing monotone* [*decreasing monotone*] on  $i$  if  $f(v_1, \dots, v_i, \dots, v_n) \leq f(v_1, \dots, v'_i, \dots, v_n)$  [ $f(v_1, \dots, v_i, \dots, v_n) \geq f(v_1, \dots, v'_i, \dots, v_n)$ ], where  $v_i = 0$  and  $v'_i = 1$ . It is *sign-definite* if it is increasing or decreasing monotone for every  $i = 1 \dots n$ . A Boolean logical model  $(V, F)$  is *sign-definite* if every Boolean function in  $F$  is sign-definite.

It is well-known that a Boolean function is sign-definite if and only if it can be represented by a formula in disjunctive normal form in which all occurrences of a given literal are either negated or non-negated [18].

We now recall *bipolar* ADFs [19].

**Definition 11.** Given a Boolean function  $f : \{0, 1\}^n \mapsto \{0, 1\}$ :

- $i \leq n$  is *supporting* iff  $f(v_1, \dots, v_i, \dots, v_n) = 1$  implies  $f(v_1, \dots, v'_i, \dots, v_n) = 1$  where  $v_i = 0$  and  $v'_i = 1$ ,
- $i \leq n$  is *attacking* iff  $f(v_1, \dots, v_i, \dots, v_n) = 0$  implies  $f(v_1, \dots, v'_i, \dots, v_n) = 0$  where  $v_i = 0$  and  $v'_i = 1$ .

An ADF is *bipolar* iff for every  $a \in \text{At}$ , every  $b \in \text{par}_D(a)$  is supporting, attacking or both in  $C_a$ .

It was shown that an ADF is bipolar iff every acceptance formula is equivalent to a formula that is syntactically bipolar, i.e., no atoms occurs both positively and negatively [20]. From this, we can show that these two special cases coincide:

**Corollary 1.** A Boolean model is sign-definite iff the corresponding ADF is bipolar.

## 4. Insights from Boolean Networks

Based on the established correspondence, we provide examples of insights from the literature on Boolean Networks that are directly relevant for ADFs.

### 4.1. Complexity of Counting Problems

For many applications in KR, being able to know the number of solutions is important. For many formalisms, including abstract argumentation [21], according decision problems have been studied in depth. For ADFs, such a study is missing. Due to existing results from the literature on BNs we can start filling this gap.

For example, Bridoux et al. [22] study the complexity of deciding whether the number of stable states in a BN is above (or below) a given bound  $k$ . Given the correspondence between BNs and ADFs established here, we immediately obtain the following corresponding results:

**Corollary 2.** Deciding whether the maximal number of two-valued models in an ADF  $D$  is larger or equal than  $k$  is: in P for fixed  $k = 1$ ; NP-complete for fixed  $k \geq 2$ ; NEXPTIME-complete for arbitrary  $k$ ; and  $\text{NP}^{\#P}$ -complete for an arbitrary  $k$  if the maximum in-degree of  $D$  is bounded by a constant  $d \geq 2$ .

Deciding whether the minimal number of two-valued models in an ADF  $D$  is smaller than  $k$  is: NEXPTIME-complete for arbitrary  $k$  without any bound on the in-degree;  $\text{NP}^{\text{NP}}$ -complete for fixed  $k \geq 2$ , if the maximum in-degree of  $D$  is bounded by a constant  $d \geq 2$ ;  $\text{NP}^{\#P}$ -complete for an arbitrary  $k$  if the maximum in-degree of  $D$  is bounded by a constant  $d \geq 2$ .<sup>2</sup>

## 4.2. Existence of Fixpoints

There is a large line of work in the literature on Boolean networks that studies structural properties of Boolean networks that affect the existence (and the number) of fixed points. Such work immediately translates to the existence of two-valued models in ADFs. E.g., several works [23, 24, 25, 26] provide a theoretical analysis of the existence of fixpoints in sign-definite BN in relation to structural parameters of the corresponding Boolean networks. We follow Aracena [26] in defining a path in a BN as *positive* if the number of negative arcs is even, and negative otherwise.

The first few results study the connection between cycles, their parity, and the existence and number of fixpoints:

**Proposition 6.** If a sign-definite BN has: no cycles, then it has a unique stable state [23]; no positive cycles, then it has at most one stable state [24]; no negative cycles, then it has at least one stable state [25]. If a BN  $N$  has a strongly connected component  $H$  such that all cycles of  $H$  are negative and for each arc  $(v_i, v_j)$  in the  $N$ ,  $v_j \in H$  then  $v_i \in H$ , then  $N$  has no stable states [26].

We derive the following corollary for ADFs:

**Corollary 3.** Let a bipolar ADF  $D$  be given. Then: if  $D$  has no cycles, then it has a unique two-valued model; if  $D$  has no positive cycles, then it has at most one two-valued model; if  $D$  has no negative cycles, then it has at least one two-valued model. If  $D$  has a strongly connected component  $H$  such that all cycles of  $H$  are negative and for each arc  $(v_i, v_j)$  in the  $N$ ,  $v_j \in H$  then  $v_i \in H$ , then  $N$  has no two-valued models.

One finds also more intricate results on the existence of fixpoints in the literature, for example in terms of so-called non-expansive maps, that guarantee the existence of stable states [27]. These fall outside the scope of this paper.

Some work has also investigated the connection between the existence of stable states and the structure of the corresponding BN:

**Proposition 7** ([26]). If a BN has at least one stable state, then it has at least one positive cycle.

We derive the following corollary for ADFs:

**Corollary 4.** An ADF has a two-valued model has at least one positive cycle.

The final result we discuss is an upper bound on the number of stable states in terms of the number of *feedback vertex sets* (FVSs). An FVS of a digraph  $G = (V, E)$  is defined to be a set of vertices that contains at least one vertex of each cycle of  $G$ . The minimum number of vertices of a FVS is denoted by  $\tau(G)$ .

**Proposition 8** ([26]). A Boolean logical model  $N = (V, F)$ , where  $|V^-(v)| \geq 1$  for all  $v \in V$ , has at most  $2^{\tau(N)}$  stable states.

This carries over to ADFs as follows.

**Corollary 5.** An ADF  $D = (\text{At}, L, C)$  s.t. every argument has at least one attacker has at most  $2^{\tau(\text{At}, L)}$  stable states.

## 5. Conclusion

We have reviewed the main syntactic similarities between ADFs and BNs, and demonstrated that these extend in large parts to the semantical level. Furthermore, we have shown the fruitfulness of this connection by deriving results on complexity and existence of semantics for ADFs based on existing results for BNs. We hope that our results will lead to further cross-contamination between these two fields, such as the use of implementations or benchmarks.

**Acknowledgments** We thank the anonymous reviewers for their valuable feedback. This work is partially supported by project BIO-REVISE (2023.13327.PEX) and NOVA LINC ref. UIDB/04516/2020 (<https://doi.org/10.54499/UIDB/04516/2020>) and ref. UIDP/04516/2020 (<https://doi.org/10.54499/UIDP/04516/2020>) with the financial support of FCT.

## References

- [1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *AI 77* (1995) 321–358.
- [2] C. Cayrol, M.-C. Lagasque-Schiex, On the acceptability of arguments in bipolar argumentation frameworks, in: *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Springer, 2005, pp. 378–389.
- [3] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, S. Woltran, Abstract dialectical frameworks revisited, in: *Procs. of ICJAI*, 2013.
- [4] S. Polberg, Understanding the abstract dialectical framework, in: *Procs. of JELIA*, volume 10021 of *LNCs*, Springer, 2016, pp. 430–446.
- [5] L. Glass, S. A. Kauffman, The logical analysis of continuous, non-linear biochemical control networks, *Journal of Theoretical Biology* 39 (1973) 103–129.
- [6] R. Thomas, Boolean formalization of genetic control circuits, *Journal of Theoretical Biology* 42 (1973) 563–585.
- [7] L. Salinas, L. Gómez, J. Aracena, Existence and non existence of limit cycles in boolean networks, in: *Automata and Complexity*, volume 42, Springer, 2022, pp. 233–252.
- [8] H. Klarner, A. Bockmayr, H. Siebert, Computing maximal and minimal trap spaces of boolean networks, *Nat. Comput.* 14 (2015) 535–544.

<sup>2</sup>The maximum in-degree of an ADF  $(\text{At}, L, C)$  is the maximal number of different literals occurring in an acceptance formula  $C_a \in C$ .

- [9] A. Naldi, E. Remy, D. Thieffry, C. Chaouiya, Dynamically consistent reduction of logical regulatory graphs, *Theor. Comput. Sci.* 412 (2011) 2207–2218.
- [10] A. Faure, A. Naldi, C. Chaouiya, D. Thieffry, Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle, *ISMB (Supplement of Bioinformatics)* 22 (2006) 124–131.
- [11] A. Garg, A. D. Cara, I. Xenarios, L. Mendoza, G. D. Micheli, Synchronous versus asynchronous modeling of gene regulatory networks, *Bioinformatics* 24 (2008) 1917–1925.
- [12] F. Gouveia, Model Revision of Boolean Logical Models of Biological Regulatory Networks, Ph.D. thesis, Instituto Superior Técnico, Universidade de Lisboa, 2021.
- [13] J. D. Schwab, S. D. Kühlwein, N. Ikonomi, M. Kühl, H. A. Kestler, Concepts in boolean network modeling: What do they all mean, *Computational and Structural Biotechnology* 18 (2020) 571–582.
- [14] M. Hopfensitz, C. Müssel, M. Maucher, H. A. Kestler, Attractors in boolean networks: a tutorial, *Comput. Stat.* 28 (2012) 19–36.
- [15] H. Klarner, A. Bockmayr, H. Siebert, Computing symbolic steady states of boolean networks, in: *Proc. of ACRI*, volume 8751 of *LNCS*, Springer, 2014, pp. 561–570.
- [16] K. Moon, K. Lee, L. Paulevé, Computational complexity of minimal trap spaces in boolean networks, *arXiv preprint arXiv:2212.12756* (2022).
- [17] J. Heyninck, G. Kern-Isberner, An epistemic interpretation of abstract dialectical argumentation, in: *Computational Models of Argument*, 2020, pp. 227–238.
- [18] M. Anthony, *Discrete mathematics of neural networks: selected topics*, SIAM, 2001.
- [19] H. Strass, Approximating operators and semantics for abstract dialectical frameworks, *Artif. Intell.* 205 (2013) 39–70.
- [20] H. Strass, Expressiveness of two-valued semantics for abstract dialectical frameworks, *Journal of Artificial Intelligence Research* 54 (2015) 193–231.
- [21] J. K. Fichte, M. Hecher, A. Meier, Counting complexity for reasoning in abstract argumentation, in: *Proc. of AAAI*, 2019, pp. 2827–2834.
- [22] F. Bridoux, A. Durbec, K. Perrot, A. Richard, Complexity of fixed point counting problems in boolean networks, *J. Comput. Syst. Sci.* 126 (2022) 138–164.
- [23] F. Robert, Iterations sur des ensembles finis et automates cellulaires contractants, *Linear Algebra and its applications* 29 (1980) 393–412.
- [24] É. Remy, P. Ruet, D. Thieffry, Graphic requirements for multistability and attractive cycles in a boolean dynamical framework, *Advances in Applied Mathematics* 41 (2008) 335–350.
- [25] A. Richard, Negative circuits and sustained oscillations in asynchronous automata networks, *Advances in Applied Mathematics* 44 (2010) 378–392.
- [26] J. Aracena, Maximum number of fixed points in regulatory boolean networks, *Bull. Math. Biol.* 70 (2008) 1398–1409.
- [27] A. Richard, Local negative circuits and fixed points in non-expansive boolean networks, *Discrete Applied Mathematics* 159 (2011) 1085–1093.

## A. Proofs for Section 3 (Boolean Networks)

**Proposition 1.** Let  $G_{STG}$  be the synchronous state transition graph of the Boolean Model  $M$  with regulatory graph  $G$ , and  $D_{M,G}$  the corresponding ADF. Then,  $\omega$  is a stable state of  $G_{STG}$  iff  $\omega$  is a two-valued model of  $D_{M,G}$ .

*Proof.* This follows from the following list of equivalences:  
 $\omega$  is a stable state for the synchronous transition graph of  $M = (V, F)$   
 $\Leftrightarrow s_i = f_i(\omega)$  for every  $s_i \in \text{At}$   
 $\Leftrightarrow \omega(s) = \omega(C_s)$  for every  $s \in \text{At}$   
 $\Leftrightarrow \omega$  is a two-valued model of  $D$ .  $\square$

**Proposition 2.** Let  $m$  be a subspace  $m$ , and  $\nu_m : V \mapsto \{0, 1, \mathbb{U}\}$  defined by

$$\nu_m(v) = \begin{cases} m(v) & \text{if } v \in D_m \\ \mathbb{U} & \text{otherwise.} \end{cases} \quad (1)$$

It holds that  $[\nu_m]^2 = S[m]$ .

*Proof.* It suffices to observe that  $\nu_m \leq_i \omega$  iff  $\omega(v) = m(v)$  for every  $v \in D_m$ .  $\square$

**Proposition 3.** Let  $M$  be a Boolean Model with regulatory graph  $G$ , and  $D_{M,G}$  the corresponding ADF:  $m$  is a trap space of  $M$  iff  $\nu_m$  is admissible in  $D_{M,G}$ .

*Proof.* We first need some preliminaries due to Klarner et al. [8]. Given a boolean function  $f$ ,  $f[m]$  is the expression obtained by substituting every occurrence of some  $v \in D_m$  in  $f$  by  $m(v)$ . For example, given  $m = 0 \star 1$  and  $f = (a \vee b) \wedge c$ ,  $f[m] = (0 \vee 1) \wedge c$ . We let  $U_m := \{v_i \in V \mid f_i[m] \text{ is constant}\}$  and define  $F[m] : V \mapsto \{0, 1, \mathbb{U}\}$  as:

$$F[m](v_i) = \begin{cases} f_i[m] & \text{if } v_i \in U_m \\ \star & \text{otherwise} \end{cases} \quad (2)$$

The core of the proof depends on the following result:

**Proposition 9** ([8], Theorem 1). A space  $m$  is a trap set if and only if  $S[m] \subseteq S[F[m]]$ .

Indeed, with Proposition 2,  $[\nu_m]^2 \subseteq [\nu_{F[m]}]^2$ , i.e.  $\nu_m \leq_i \nu_{F[m]}$ . We now show that  $\Gamma_{D_{M,G}}[\nu_m] = \nu_{F[m]}$ . Recall that  $\Gamma_{D_{M,G}}[\nu_m](v_i) = \prod_i \{\omega(C_s) \mid \omega \in [\nu_m]^2\}$ . We can safely assume that  $C_s$  is in conjunctive normal form, as  $\Gamma_{D_{M,G}}$  is invariant under classical equivalences. Let  $C_s = \bigwedge_{i=1}^n \bigvee \Delta_i$ . Then  $f_s[m] = \bigwedge_{i=1}^n \bigvee \sigma(\Delta_i)$  where  $\sigma(\Delta)$  is obtained by replacing every occurrence of some  $v \in D_m$  by  $m(v)$ . Suppose now  $F[m](s) = 0$ . This means there is some  $i = 1, \dots, n$  s.t.  $\sigma(\Delta_i) = \{0\}$ . Thus,  $\prod_i \{\omega(\bigvee \Delta_i) \mid \omega \in [\nu_m]^2\} = 0$  which implies that  $\prod_i \{\omega(\bigwedge_{i=1}^n \bigvee \Delta_i) \mid \omega \in [\nu_m]^2\} = 0$ . The cases for  $F[m](s) = 1$  and  $F[m](s) = \star$  are similar.  $\square$

**Proposition 4.** Let  $m_1, m_2$  be two subspaces. Then  $m_1 \leq m_2$  iff  $\nu_{m_2} \leq_i \nu_{m_1}$ .

*Proof.* With Proposition 2 from [17],  $\nu_{m_2} \leq_i \nu_{m_1}$  iff for every  $s \in \text{At}$  iff  $[\nu_{m_2}]^2 \supseteq [\nu_{m_1}]^2$ . As  $S[m_i] = [\nu_{m_i}]^2$  for  $i = 1, 2$  (Proposition 2), we obtain that  $\nu_{m_2} \leq_i \nu_{m_1}$  iff  $S[m_1] \subseteq S[m_2]$ . By definition of  $\leq$ ,  $\nu_{m_2} \leq_i \nu_{m_1}$  iff  $m_1 \leq m_2$ .  $\square$

**Proposition 5.** Let  $M$  be a BN with regulatory graph  $G$ , and  $D_{M,G}$  the corresponding ADF:  $m$  is a minimal trap space of  $M$  iff  $\nu_m$  is preferred in  $D_{M,G}$ .

*Proof.* With Proposition 3,  $m$  is a trap space of  $M$  iff  $\nu_m$  is admissible in  $D_{M,G}$ . Suppose now towards a contradiction that  $m$  is a minimal trap space, yet  $\nu_m$  is not preferred, i.e. there is some admissible  $\nu$  with  $\nu_m <_i \nu$ . Then with Proposition 4 and Proposition 3, there is a trap space  $m_\nu < m$ , contradicting  $m$  being a minimal trap space.  $\square$

The reader might be somewhat surprised by the fact that links can be both attacking and supporting. However, the only possibility for that being the case, is if the link is redundant:

**Proposition 10.** Consider an ADF  $D = (\text{At}, L, C)$  with  $s_1, s_2 \in \text{At}$  and  $s_2$  being supporting and attacking in  $C_{s_1}$ . For every  $v_1, v_2 \in \mathcal{V}^2(\text{At})$  s.t.  $v_1(s_2) \neq v_2(s_2)$  and  $v_1(s) = v_2(s)$  for every  $s \in \text{At}$ ,  $\Gamma_D(v_1)(s_1) = \Gamma_D(v_2)(s_1)$ .

**Corollary 1.** A Boolean model is sign-definite iff the corresponding ADF is bipolar.

*Proof.* We first recall that, given a formula  $\phi$ , the *polarity* of an atom  $a$  in  $\phi$  is determined by the number of negations on the path from the root of the formula tree to the atom, and is positive if this number is even, and negative otherwise. For example, in  $\neg(a \wedge \neg b)$ ,  $a$  occurs negatively and  $b$  occurs positively. A propositional formula is *syntactically bipolar* if no atom  $a$  occurs both positively and negatively in  $\phi$ .

Clearly, an acceptance condition is supporting respectively attacking iff it is increasing respectively decreasing monotone. The rest of the proof is an immediate consequence of [20, Theorem 1], which implies that an ADF  $D = (\text{At}, L, C)$  is bipolar iff, for every  $s \in \text{At}$ ,  $C_s$  is syntactically bipolar.  $\square$