Computational Models of Argument
C. Reed et al. (Eds.)

doi:10.3233/FAIA240312

Connecting Abstract Argumentation and Boolean Networks

Yannis DIMOPOULOS [a], Wolfgang DVOŘÁK [b] and Matthias KÖNIG [b,1]
[a] *University of Cyprus, Department of Computer Science*
[b] *TU Wien, Institute of Logic and Computation*

Abstract. Already in Dung's seminal paper introducing Abstract Argumentation Frameworks (AFs), several connections to seemingly unrelated reasoning formalisms have been illustrated. In this work, we continue this trend and establish a connection between abstract argumentation frameworks and boolean networks (BNs). BNs, in a nutshell, mimic simple binary-valued systems, where for each point in time, the value of each bit (component) depends only on the other components' values of the previous point in time of the network. This formalism is widely used to formally analyze biological processes, where from simple rules complex behavior emerges. We show that stable extensions of an arbitrary AF correspond to single state attractors of its canonically corresponding BN, the complete extensions correspond to a distinctive 2-state attractor, and the admissible sets correspond to the seeds of the BN. We thereby lay the groundwork for a fruitful exchange of ideas between the two research areas.

Keywords. Abstract Argumentation, Boolean Networks

1. Introduction

In this work we investigate the relationship between Dung's abstract argumentation frameworks [1] and Boolean Networks. *Boolean Networks* (BN) [2] were introduced by S. Kauffman [3] and R. Thomas [4] for representing gene regulatory networks. They gained wide popularity in the early 2000s as a qualitative approach to the structural analysis and dynamic modeling of biological systems. Their importance was further strengthened with the advent of system biology that investigates the ways the interactions between the components of a biological system give rise to the system's behavior.

In a Boolean network model each node can assume one of the two possible values True (ON) or False (OFF). This value represents the state of that node, for instance whether a gene is expressed or not. The future state of a node is determined by a boolean function on the current states of its regulators. A BN eventually reaches a set of stable or steady states, i.e. a cyclic sequence of states that the system visits successively and repeatedly. These sets are called *attractors*. Attractors represent the long-term behavior of a BN and have been linked to biological phenotypes (physical properties of an organism), attracting for this reason a large body of research (see, e.g., [5,6,2]).

[1]Corresponding Author: Matthias König, matthias.koenig@tuwien.ac.at

In this paper we establish the relation between the semantics of AFs and the attractors of their corresponding BNs. It turns out that argumentation frameworks can be represented by a family of BNs with boolean functions that are conjunctions of negative literals. The single state attractors of these BNs are exactly the stable extensions of the corresponding AF. Moreover, it is shown that the complete extensions correspond to 2-state attractors of the associated BN, whereas admissible extensions correspond to seeds, that are roughly sets of states in the BN that once reached cannot be escaped.

As a first result of our work, one can use argumentation solvers to obtain some attractors of BNs of this form without enumerating the exponentially large state space.

2. Background

2.1. Abstract Argumentation

In this section we briefly recall Argumentation Frameworks (AFs) due to Dung [1].

Definition 1. *An argumentation framework is a pair $F = (A,R)$ where A is a non-empty of arguments, and $R \subseteq A \times A$ is the attack relation. Let $S \subseteq A$ be a set of arguments. We say a set S* attacks *an argument $a \in A$ if $(b,a) \in R$ for some $b \in A$. We denote by S^+ the set $\{a \in A \mid S \text{ attacks } a\}$.*

The semantics of AFs are defined via *extensions*, i.e., jointly acceptable sets of arguments. In this work we focus on complete and stable extensions, and the underlying notion of admissibility.

Definition 2. *Let $F = (A,R)$ be an AF, a set $S \subseteq A$ is* conflicting *in F if S attacks a for some $a \in S$, otherwise S is* conflict-free. *An argument $a \in A$ is* defended *(in F) by $S \subseteq A$ if for each $(b,a) \in R$ it holds that S attacks b. Let $S \subseteq A$ be a conflict-free set in F. Then,*

- *S is* admissible *in F, denoted by $S \in adm(F)$, if S defends each $a \in S$ in F;*
- *S is* stable *in F, denoted by $S \in stb(F)$, if S attacks every argument in $A \setminus S$; and*
- *S is* complete *for F, denoted by $S \in com(F)$, if $S \in adm(F)$ and contains every argument it defends.*

2.2. Boolean Networks

This section presents the basic concepts of boolean networks related to the purposes of this paper, following the notation of [6].

A *Boolean Network* is defined as a pair $N = (V,F)$ where $V = \{x_1,\dots,x_n\}$ is a set of boolean variables, and $F = \{f_1,\dots,f_n\}$ is a set of corresponding boolean functions. In the context of biological systems, the value $x_i = 1$ (ON) encodes that the entity represented by variable x_i, i.e., gene, protein, or molecule, is active or expressed or is above a certain concentration threshold, while the value $x_i = 0$ (OFF) represents that the entity is inactive, not expressed, or is below a certain concentration threshold [7].

Boolean networks are used to represent discrete dynamic systems, where at each time t, each variable $x_i \in V$ has an associated *state* $x_i(t) \in \{0,1\}$. A value assignment to all variables of network N, denoted by $s_N(t)$, represents the *network state* at time t, composed of n individual variable states, i.e., $s_N(t) = (x_1(t),\dots,x_n(t))$. The set $S = S(V)$

of all 2^n states of N is its *state space*. To disburden notation, we omit the Boolean network N, and write $s(t)$ instead of $s_N(t)$. Moreover, when the specific time t is not relevant, we write $s = (x_1,\ldots,x_n)$ to denote $s(t) = (x_1(t),\ldots,x_n(t))$. Finally, for a state $s = (x_1,\ldots,x_n)$ we slightly abuse the notation and write $s(x_i)$ to obtain the value of s for variable x_i.

A Boolean function $f_i(x_{i_1},\ldots,x_{i_k})$, $f_i \in F$, called the *update* or *transition* function, is associated with each variable $x_i \in X$, specifying that the value of x_i is determined by the values of the nodes of the set $IN(x_i) = \{x_{i_1},\ldots,x_{i_k}\}$. The state of variable x_i at time $t+1$ is $x_i(t+1) = f_i(x_{i_1}(t),\ldots,x_{i_k}(t))$ which we simplify to $x_i(t+1) = f_i(x_i(t),\ldots,x_n(t))$. The interaction between the variables of a boolean network N is captured by the *interaction graph* of N, defined as $G_N = (V,E)$, with $E = \{(x_{i_j},x_i) \mid x_{i_j} \in IN(x_i)\}$.

The order in which the variable updates are carried out in a BN is determined by its *update method*, which can be *synchronous* or *asynchronous*. In synchronous BNs, the states of all variables are updated *simultaneously*, whereas in *asynchronous* BNs the variable states are not updated concurrently, but *separately* [8]. In this work we focus on synchronous Boolean Networks. For these networks, a transition step is performed on a state $s = (x_1,\ldots,x_n)$ by applying each local transition function simultaneously, and obtaining $F(s) = (f_1(s),\ldots,f_n(s))$ as the successor state, i.e. $s(t+1) = (x_1(s(t)),\ldots,x_n(s(t)))$.

Naturally, the transition relation can be interpreted as a directed graph.

Definition 3 (Transition Graph)**.** *Let $N = (V,F)$ be a boolean network, with $V = \{x_1,\ldots,x_n\}$. The* transition graph $\Sigma(N)$ *is the directed graph with the* 2^n *states* $\{0,1\}^n$ *as vertices and an edge* (s_1,s_2) *iff* $s_2 = F(s_1)$*. We call these transitions between states the (transition-)* steps*.*

It is clear that each state has exactly one successor state. Hence, applying several transitions steps (i.e., following a "transition path") in a final boolean network f inevitably leads to a cycle in $\Sigma(N)$. These cycles are the *attractors* of N.

Definition 4 (Attractors)**.** *Let N be a boolean network and $\Sigma(N)$ its transition graph. An* attractor *of N is each sequence of states $s_1,\ldots,s_m$ such that for each $2 \leq i \leq m$ it holds $s_i = F(s_{i-1})$ and $s_1 = F(s_m)$. An attractor $s_1,\ldots,s_m$ is* single state *or* singleton *if $m = 1$ and* cyclic *or* periodic *if $m \geq 2$. In the latter case, the* period *of the attractor is m.*

In other words, each strongly connected component of $\Sigma(N)$ forms an attractor.

Another family of dynamically closed subspaces, i.e. subsets of the state space that no sequence of transitions can escape, are the *symbolic steady states* and the *seeds*, introduced in [9,10]. They both hinge on 3-valued states, called *partial states*.

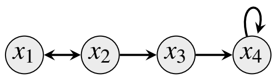
Definition 5 (Partial state)**.** *Let $N = (V,F)$ be a Boolean Network, with $V = \{x_1,\ldots,x_n\}$ and $F = \{f_1,\ldots,f_n\}$. A* partial state *s of N is an assignment $(x_1,\ldots,x_n) \in \{0,1,\theta\}^n$. The set of variables $D_s = \{x \mid s(x) \neq \theta\}$ is the* domain *of s. A state $s' = (x_1,\ldots,x_n) \in \{0,1\}^n$* extends *a partial state s if $s'(x) = s(x)$ for all $x \in D_s$. The set $\mathrm{X}(s)$ contains all states that extend the partial state s. Two partial states s,s' are* compatible *if for each variable $x \in D_s \cap D_{s'}$ it holds that $s(x) = s'(x)$.*

Given an expression r and a partial state s, the expression $r[s]$ is obtained by substituting in r the values $s(x)$ for all $x \in D_s$. For instance, given $r = x_1 \wedge x_2$ and $s = (1,\theta)$, then $r[s] = 1 \wedge x_2 = x_2$. The image $F(s)$ of a partial state s w.r.t. a Boolean Network $N = (V,F)$ is the partial state s' with $D_{s'} = \{x_i \in V \mid f_i[s]$ is constant$\}$ and $s'(x_i) = f_i[s]$, for all $x_i \in D_{s'}$.

Definition 6 (Seeds and Symbolic Steady States)**.** *A partial state s is a* seed *for a Boolean Network $N = (V,F)$ if s and $F(s)$ are compatible and $D_s \subseteq D_{F(s)}$; a seed is a* symbolic steady state *if $D_s = D_{F(s)}$.*

In [10], the notion of a trap set is introduced. A non-empty set of states $R \subseteq S$ is a *trap set* if for every $r \in R$ it holds that $F(r) \in R$. The inclusion-wise minimal trap sets of a boolean network N correspond to the attractors of N. It is noted that every trap set contains at least one attractor.
In [10] it is shown that if a partial state s is a seed of a boolean network N, then the set $\mathrm{X}(s)$ of states extending s is a trap set for N.

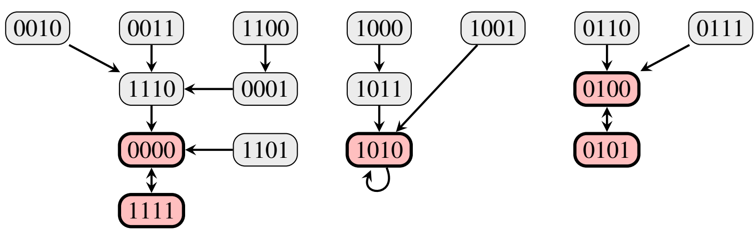
Example 7. *Consider the Boolean network $N = (V,F)$ with $V = \{x_1,x_2,x_3,x_4\}$, and F as depicted below, together with its interaction graph.*



$f_1 = \neg x_2 \quad f_3 = \neg x_2$
$f_2 = \neg x_1 \quad f_4 = \neg x_3 \wedge \neg x_4$

The seeds of N are the partial states: $s_1 = (\theta,\theta,\theta,\theta)$ (because $F(s_1) = (\theta,\theta,\theta,\theta)$), $s_2 = (1,0,\theta,\theta)$ ($F(s_2) = (1,0,1,\theta)$), $s_3 = (0,1,\theta,\theta)$ ($F(s_3) = (0,1,0,\theta)$), $s_4 = (1,0,1,\theta)$ ($F(s_4) = (1,0,1,0)$), $s_5 = (0,1,0,\theta)$ ($F(s_5) = (0,1,0,\theta)$), $s_6 = (1,0,1,0)$ ($F(s_6) = (1,0,1,0)$).

The transition graph $\Sigma(N)$ is depicted below. The states are represented as binary strings rather than tuples, e.g. the tuple $(0,1,1,0)$ is represented by the string 0110, where 0 is the value of the first variable x_1 of V, 1 the value of the second variable x_2 etc.



The seed $s_2 = (1,0,\theta,\theta)$ induces the trap set $R_2 = \{1000,1001,1011,1010\}$, seed $s_4 = (1,0,1,\theta)$ the set $R_4 = \{1011,1010\} \subseteq R_2$, and finally $s_6 = (1,0,1,0)$ the trap set $R_6 = \{1010\} \subseteq R_4$. Moreover, the minimal trap set R_6 is a singleton attractor. Other attractors of the above network are the sets $\{0000,1111\}$ and $\{0100,0101\}$, both marked in red in the figure.

3. Stable Extensions and Single State Attractors

A concise definition of Boolean Networks, that is suitable for addressing networks characterized by complete (i.e., non-partial) states, is the following ([11]). It reduces boolean networks to their corresponding global transition function.

Definition 8 (Boolean Network)**.** *A* boolean network *on n variables or components is described by its* transition function $f : \{0,1\}^n \to \{0,1\}^n$ *where a* state $s = (x_1,\ldots,x_n) \in \{0,1\}^n$ *maps to $f(s)$ via n* local transition functions $f(s) = (f_1(s),\ldots,f_n(s))$.

We can model an argumentation framework (A,R) as a boolean network where each component models an argument $a \in A$. The intuition is that each state $s = (a_1,\ldots,a_n)$ corresponds to a subset $S \subseteq A = \{a_1,\ldots,a_n\}$ of arguments, in that $a_i = 1$ in s iff $a_i \in S$ (and $a_i = 0$ if $a_i \notin S$). It remains to define a suitable transition function. For this, we recall that in stable extensions each argument is in an extension iff all of its attackers are not in the extension. Hence, we obtain the following local transition functions.

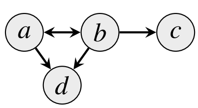
Definition 9 (Abstract Argumentation as Boolean Network)**.** *Let $T = (A,R)$ be an AF. Its corresponding boolean network f^T consists of $n = |A|$ variables, with*

$$f_a = \bigwedge_{(b,a)\in R} \neg b$$

For a state $s = (a_1,\ldots,a_n) \in \{0,1\}^n$ we define $\{a \mid s(a) = 1\}$ as its corresponding set of arguments, denoted by A_s. Likewise, for a set of arguments $M \subseteq A$ we define the state s s.t. $s(a) = 1$ iff $a \in M$ as its corresponding state, identified by s_M.

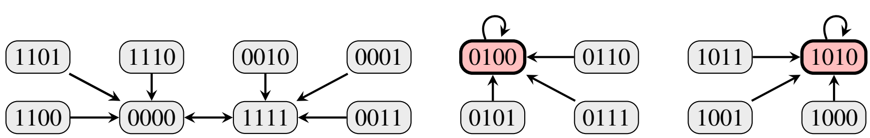
Notice that, by the above, also each boolean network where all transition functions are conjunctions of negative literals naturally corresponds to an AF. That is, we have a one-to-one mapping between AFs and this class of boolean networks.

Example 10. *Consider the following AF T and its corresponding boolean network f^T.*



$$f_a^T = \neg b \qquad f_c^T = \neg b$$
$$f_b^T = \neg a \qquad f_d^T = \neg a \wedge \neg b$$

The AF T also acts as interaction graph for the boolean network f^T. We have $stb(T) = \{\{b\},\{a,c\}\}$. We obtain the following transition graph $\Sigma(f^T)$.



We have two single state attractors (highlighted):

1. $(0,1,0,0)$ *corresponding to* $\{b\}$*, and*
2. $(1,0,1,0)$ *corresponding to* $\{a,c\}$.

As we have seen, in Example 10 the single state attractors exactly correspond to the stable extensions of the corresponding argumentation framework. This is no coincidence, as we show in our first main result.

Theorem 11. *Let $T = (A,R)$ be an AF. The single state attractors of f^T are in a one-to-one correspondence with the stable extensions of T.*

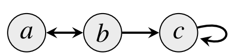
Proof. First assume $s = (a_1,\ldots,a_n)$ is a single state attractor of f^T and A_s is its corresponding set of arguments. This means, by the definition of f^T, that for each a with $s(a) = 1$ we have that for each attack $(b,a) \in R$ towards the corresponding argument, it holds $s(b) = 0$ (otherwise the value of a would change to 0 in the next step). Hence, A_s is conflict-free in T. Moreover, for each a with $s(a) = 0$ we have that for at least one b where $(b,a) \in R$ it holds $s(b) = 1$ (otherwise the value of a would change to 1 in the next step). Hence, each argument $a \in A \setminus A_s$ is attacked by A_s, i.e., A_s is stable in T.

Now assume M is a stable extension of T and s_M is its corresponding state of f^T. This means for each argument $a \in M$, each argument b where $(b,a) \in R$ is not in M, i.e., $s_M(b) = 0$. Therefore, $s_M(a) = 1$ will not change in the next step. Likewise, for each argument $a \in A \setminus M$, since M is stable, there is an argument $b \in M$ with $(b,a) \in R$. Hence, since $s_M(b) = 1$, in the next step we will maintain $s_M(a) = 0$. In summary, s_M is a single state attractor of f^T. □

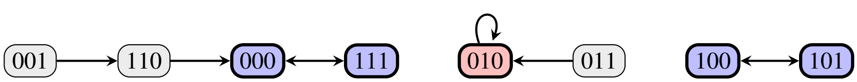
4. Characterizing Complete Extensions

Intuitively, the transition steps in the boolean network corresponding to an argumentation framework behave similar to an application of the *characteristic function*[2] of the framework. In particular, arguments that are attacked only by arguments that are not in the characterized set will be in the characterized set in the next step. However, the behavior of the boolean network is clearly not monotonic (in contrast to the characteristic function of AFs) and makes no distinction between defeated arguments (i.e., attacked by the characterized set) and *undecided* arguments (i.e., neither in nor attacked by the set). Still, we will show that there is a connection between the attractors of the boolean network and the underlying AF.

Example 12. *Consider the following AF T.*



We have $stb(T) = \{b\}$ and $com(T) = \{\emptyset, \{a\}, \{b\}\}$. We obtain the following transition graph $\Sigma(f^T)$ of the corresponding boolean network f^T.



We have three attractors (highlighted):

1. $(0,1,0)$ *corresponding to the stable extension* $\{b\}$*,*
2. $(0,0,0),(1,1,1)$ *corresponding to the complete extension* $\emptyset$*, and*
3. $(1,0,0),(1,0,1)$ *corresponding to the complete extension* $\{a\}$*.*

By Theorem 11 we get that the single state attractor $(0,1,0)$ corresponds to the stable extension $\{b\}$, for the cyclic attractors we are going to establish the correlation next.

Let us investigate the cyclic attractor $(1,0,0),(1,0,1)$. Note that we observe a 3-valued behavior: while the first component remain invariably at 1 in all states of the attractor, the second component similarly remains at 0. The third component on the other hand "oscillates" between 0 and 1. We will establish that this corresponds to a complete extension E where the argument corresponding to the first component (a in our case)

[2]For an AF $T = (A,R)$ the characteristic function Γ_T for a set $S \subseteq A$ is defined as $\Gamma_T(S) = \{a \in A \mid S \text{ defends } a\}$.

is *in* E, b is *attacked by* E, and c is neither in nor attacked by E (i.e., undecided). The intuitive reason is that all attackers of a are set to 0 and at least one attacker of b is set to 1 (namely a). However, the argument c is attacked by b and c. In the first state of the attractor $(1,0,0)$, this means that c is attacked only by arguments that are 0, and will hence be set to 1 in the next step. In the second state $(1,0,1)$ the argument c is attacked by an argument that is set to 1 (namely c itself), and will be set to 0 in the next step. Indeed, this generalizes to every complete extension and their undecided arguments, as the next result illustrates. For this, we define the following pair of states.

Definition 13 (Complete State-Pair)**.** *Let $T = (A,R)$ be an AF and let $E \subseteq A$. The following two states s, s' of f^T are the* complete state-pair w.r.t. E.

$$s(a) = \begin{cases} 1 & \textit{if } a \in E \\ 0 & \textit{if } a \in E^+ \\ 0 & \textit{if } a \in A \setminus (E \cup E^+) \end{cases} \qquad s'(a) = \begin{cases} 1 & \textit{if } a \in E \\ 0 & \textit{if } a \in E^+ \\ 1 & \textit{if } a \in A \setminus (E \cup E^+) \end{cases}$$

If a complete state-pair is an attractor, we call it a complete state-pair-attractor.

We next show how complete extensions translate to complete state-pairs.

Proposition 14. *Let $T = (A,R)$ be an AF, let $E \subseteq A$ be a complete extension, and let f^T be the corresponding boolean network. Then either*

1. *s_E is a single state attractor iff E is stable, or otherwise*
2. *the complete state-pair s, s' w.r.t. E is an attractor.*

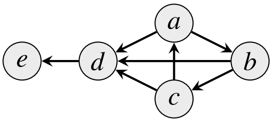
Proof. 1. follows directly from Theorem 11. For 2. assume E is a complete extension of T. For s, let us first look at the components (arguments) a with $s(a) = 1$ ($a \in E$). Since E is conflict-free, we have for each $(b,a) \in R$ that $b \notin E$ and hence $s(b) = 0$. Hence, $f_a^T(s) = 1$, i.e., a remains at 1. Now for $a \in E^+$, we have $s(a) = 0$. From $a \in E^+$ we get that there is a b with $(b,a) \in R$ s.t. $s(b) = 1$, which means $f_a^T(s) = 0$, i.e., a remains at 0. Finally, for $a \in A \setminus (E \cup E^+)$, we again have $s(a) = 0$. If a is unattacked, then $a \in E$ (since E is complete), so there is at least one $(b,a) \in R$ towards a. Since $a \notin E^+$, for all $(b,a) \in R$ it holds $s(b) = 0$, which means $f_a(s) = 1$, i.e., a is set to 1 in the successor state. We see that the successor state of s is exactly s'.

Now for s' we again first consider the components a corresponding to $a \in E$. For the same reason as for s we have that $f_a^T(s') = 1$. Likewise, let $a \in E^+$ and we get as with s that $f_a^T(s') = 0$. Now let $a \in A \setminus (E \cup E^+)$, which means $s'(a) = 1$. Let b s.t. $(b,a) \in R$. If $b \in E$ we would have $a \in E^+$, so we get $b \notin E$. If each b with $(b,a) \in R$ is in E^+, we would have $a \in E$ (since E is complete), i.e., there is at least one b s.t. $b \in A \setminus (E \cup E^+)$. For this b it holds $s'(b) = 1$, which gives us $f_a^T(s') = 0$, i.e., a is set to 0 in the next step. We now see that the successor state of s' is exactly s. Since $f^T(s) = s'$ and $f^T(s') = s$ we know s, s' is a cyclic attractor. □

The next natural question is whether *each* attractor of f^T corresponds to a complete extension. However, the following counter example[3] illustrates the opposite.

[3] This counter example was originally proposed by Loizos Michael of the Open University of Cyprus.

Example 15. *Consider the following AF T and its corresponding boolean network f^T.*



$$f_a^T = \neg c \qquad f_d^T = \neg a \wedge \neg b \wedge \neg c$$
$$f_b^T = \neg a \qquad f_e^T = \neg d$$
$$f_c^T = \neg b$$

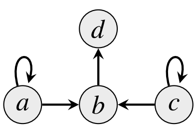
It holds $com(T) = \{\emptyset\}$. We have two attractors for f^T:

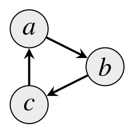
1. $(0,0,0,0,0)$, $(1,1,1,1,1)$ *which by Proposition 14 corresponds to the empty complete extension, and*
2. $(1,0,0,0,1)$, $(1,0,1,0,1)$, $(0,0,1,0,1)$, $(0,1,1,0,1)$, $(0,1,0,0,1)$, $(1,1,0,0,1)$ *which* does not *correspond to* any *complete extension.*

In the second attractor in each state component d is "deactivated" by at least one of a, b, or c. However, these arguments are not invariably set to 1, and would not be interpreted as part of a complete extension. On the other hand, component d is 0 and e is 1 in each state. Clearly, the set of arguments $\{e\}$ is not complete.

It now seems like all attractors of size 2 correspond to complete extensions, while no attractor of size ≥ 3 does. Both of these ideas are false.

Example 16. *Consider the following AFs T_1 (left) and T_2 (right) and below their respective attractors.*





1. $(0,0,0,0),(1,1,1,1)$
2. $(1,0,0,1),(0,0,1,1)$

1. $(0,0,0),(1,1,1)$
2. $(0,0,1),(0,1,1),(0,1,0),(1,1,0),(1,0,0),(1,0,1)$

Again we consider the sets of arguments corresponding to the components that are set to 1 in all states of an attractor. While in T_1 the second attractor of size 2 corresponds to $\{d\}$—which is clearly not complete, in T_2 the second attractor of size 6 corresponds to the empty set—which is indeed complete.

All hope is not lost. The next result finally provides a tool to distinguish attractors that correspond to complete extensions from those that do not. Moreover, we exactly characterize the attractors that "most closely" correspond to the complete extensions. We thereby make use of the idea of Proposition 14 and show that *each* of these state-pair-attractors correspond to a complete extension. To recall, we consider 2-state attractors, where the first state s is the one corresponding to the complete extension E in the sense of Definition 9. The second state s' coincides with s on all components that correspond to arguments in $(E \cup E^+)$, and sets the remaining components (corresponding to arguments in $A \setminus (E \cup E^+)$) to 1. For example, recalling Example 12 we have for $E = \{a\}$ that $a \in E$, $b \in E^+$, and $c \in A \setminus (E \cup E^+)$. We obtain $s = (1,0,0)$ and $s' = (1,0,1)$.

Proposition 17. *Let $T = (A,R)$ be an AF, let $M \subseteq A$ and let s_M, s'_M be the complete state-pair w.r.t. M. If s_M, s'_M form an attractor of f^T then M is a complete extension of T.*

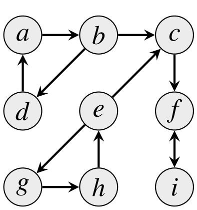
Proof. Assume $s_M \neq s'_M$ is an attractor of f^T. We have $M = \{a \in A \mid s_M(a) = 1\}$. Let a be an arbitrary argument in M, note that both in state s_M and in state s'_M it holds for all $(b,a) \in R$ that $s_M(b) = s'_M(b) = 0$, as otherwise a would be set to 0 in the next step. Hence, M is conflict-free in T. Moreover, since $s_M(b) = s'_M(b) = 0$ this means that both in s_M and s'_M there is a $(c,b) \in R$ s.t. $s_M(c) = s'_M(c) = 1$ (as otherwise b would be set to 1 in the next step). Hence, M defends itself in T. In fact, by the same reasoning we obtain that for each argument $b \in M^+$ we get $s_M(b) = s'_M(b) = 0$. Let a be an arbitrary argument that is attacked only by arguments $b \in M^+$ (i.e., an arbitrary argument that is defended by M). Since for these arguments it holds $s_M(b) = s'_M(b) = 0$, we must have $s_M(a) = s'_M(a) = 1$, i.e., M contains every argument it defends. In summary, we obtain that M is complete in T. □

Effectively, these results do not only characterize the attractors corresponding to complete extensions, but also give us a tool to check if an *arbitrary* attractor also corresponds to a complete extension (like attractor 2. in Example 16 (right)). If we have an attractor $s_1, \dots, s_m$ we observe a three-valued behavior of the components: while some components stay at 1 or 0 for all states, other components will change. If the attractor characterizes a complete extension, the components a with $s_1(a) = \dots = s_m(a) = 1$ correspond to an argument in M, the components b with $s_1(b) = \dots = s_m(b) = 0$ correspond to an argument in M^+, and the arguments c with $s_i(c) \neq s_j(c)$ for some $1 \leq i, j \leq m$ correspond to an argument in $A \setminus (M \cup M^+)$ (an undecided argument). By Proposition 14 we know that if M is complete, the corresponding state $s_M = \{a \in A \mid s_1(a) = \dots = s_m(a) = 1\}$ will form an attractor in a complete state-pair. Hence, to check if our arbitrary attractor $s_1, \dots, s_m$ corresponds to a complete extension, we can perform the following steps:

1. Identify the (potentially) characterized set $M = \{a \in A \mid s_1(a) = \dots = s_m(a) = 1\}$,
2. Compute the complete state-pair s_M, s'_M w.r.t. M (i.e., compute s'_M by setting $s'_M(c) = 1$ for the arguments c with $s_{M,i}(c) \neq s_{M,j}(c)$ for some $1 \leq i, j \leq m$), and
3. Check if $f(s_M) = s'_M$ and $f(s'_M) = s_M$, i.e., apply two steps of the transition function to check if we found a complete state-pair-attractor.

If this check is true, we know M is a complete extensions that corresponds to our (arbitrary) attractor, otherwise by Proposition 17 we know that M is not complete in T. We illustrate this in the following example.

Example 18. *Assume we put for the following AF T its corresponding boolean network f^T into a solver that outputs at first attractor 1.*



1. $(0,0,0,1,1,1,0,1,0)$,
$(0,1,0,1,0,1,0,1,0)$,
$(0,1,0,0,0,1,1,1,0)$,
$(1,1,0,0,0,1,1,0,0)$,
$(1,0,0,0,1,1,1,0,0)$,
$(1,0,0,1,1,1,0,0,0)$

3. $(0,1,0,1,0,0,0,0,1)$,
$(0,1,0,0,1,0,1,1,1)$,
$(1,1,0,0,0,0,0,0,1)$,
$(1,0,0,0,1,0,1,1,1)$,
$(1,0,0,1,0,0,0,0,1)$,
$(0,0,1,1,1,0,1,1,1)$

2. $(0,0,0,0,0,1,0,0,0)$,
$(1,1,1,1,1,1,1,1,0)$

4. $(0,0,0,0,0,0,0,0,1)$,
$(1,1,1,1,1,0,1,1,1)$

We want to find out if the attractor 1. corresponds to a complete extension, so we compute check if 2. is an attractor. We have $f^T((0,0,0,0,0,1,0,0,0)) = (1,1,1,1,1,1,1,1,0)$.

However, we get $f^T((1,1,1,1,1,1,1,1,0)) = (0,0,0,0,0,0,0,0,0)$*, i.e., we do not have an attractor. We can conclude that the corresponding set* $\{f\}$ *is not complete (and indeed, it can easily be verified that* $\{f\}$ *is not defended against the attack from c).*

Now assume our solver provides attractor 3. We again check whether 4. is an attractor, and indeed this time we are lucky. Indeed, the corresponding set $\{i\}$ *is complete.*

Finally, combining Proposition 14 and Proposition 17 we immediately obtain the following characterization for complete extensions.

Theorem 19. *Let* $T = (A,R)$ *be an AF. The complete state-pair-attractors of* f^T *correspond to the complete extensions of* T*.*

First note that this result subsumes the result from Theorem 11 (as all stable extension are also complete) if we allow the slight abuse of notation where a single state attractor s can be written as a complete state-pair s,s. Also note that the "correspondence" in Theorem 19 should be interpreted slightly differently than in Theorem 11, where the single state attractor in question has a one-to-one correspondence to the complete extensions in the sense of Definition 9. This is not the case for Theorem 19 as we saw in Example 18 where the complete extension $\{i\}$ corresponds both to attractor 3. and 4. However, there is a one-to-one correspondence between complete extensions and complete state-pair-attractors.

5. Seeds and Symbolic Steady States

In this section we consider partial states of boolean networks and investigate how they relate to admissible sets and complete extensions. First we show that admissible sets correspond to seeds in the corresponding boolean network.

Theorem 20. *Let* $T = (A,R)$ *be an AF and let* f^T *be its corresponding boolean network. A set* $S \subseteq A$ *is admissible in* T *iff* s *is a seed in* f^T *with*

$$s(a) = \begin{cases} 1 & \text{if } a \in S \\ 0 & \text{if } a \in S^+ \\ \theta & \text{otherwise} \end{cases}$$

Proof. With a slight abuse of notation, we will use $f^T(s)$ to refer to the image of a partial state s.

$\Rightarrow$: Let $S \subseteq A$ be admissible in T. Let $a \in S$, we have $s(a) = 1$ and $s(b) = 0$ for each b where $(b,a) \in R$, which in turn means $f^T(s')(a) = 1$ for each state s' extending s. Now let $a \in S^+$, we have $s(a) = 0$ and $s(b) = 1$ for some b where $(b,a) \in R$, which in turn means $f^T(s')(a) = 0$ for each state s' extending s. Hence, s is a seed of f^T.

$\Leftarrow$: Let s be a seed in f^T. Let a be s.t. $s(a) = 1$. We have for each b s.t. $(b,a) \in R$ that $s(b) = 0$, as otherwise there is a state s' extending s with $s'(b) = 1$ where $f^T(s')(a) = 0$ which would mean that s is not a seed, a contradiction. This means the corresponding set $S = \{a \in A \mid s(a) = 1\}$ is conflict-free. Moreover, this means there is a c s.t. $(c,b) \in R$ with $s(c) = 1$, since otherwise there is a state s' extending s where for each such c it holds $s(c) = 0$ which means $f^T(s')(b) = 1$ which in turn means $f^T(f^T(s'))(a) = 0$ which would again mean that s is not a seed, a contradiction. This means that c defends a in T, and, hence, S defends itself in T, i.e., S is admissible. □

Notice however, that this correspondence is many-to-one in the sense that several seeds can correspond to the same admissible set. This mirrors the relation between admissible sets and admissible labellings discussed in [12], where several labelings that agree on the accepted arguments but disagree on which arguments are labeled out, or undecided respectively, correspond to the same admissible set.

Next we show a one-to-one correspondence between symbolic steady states of BNs and complete extensions of the corresponding AF.

Theorem 21. *Let $T = (A,R)$ be an AF and let f^T be its corresponding boolean network. A set $S \subseteq A$ is complete in T iff s is a symbolic steady state in f^T with*

$$s(a) = \begin{cases} 1 & \textit{if } a \in S \\ 0 & \textit{if } a \in S^+ \\ \theta & \textit{otherwise} \end{cases}$$

Proof. With a slight abuse of notation, we will use $f^T(s)$ to refer to the image of a partial state s.

$\Rightarrow$: Assume S is complete. By Theorem 20 we get that s is a seed. Assume towards contradiction that $D_S \subset D_{f^T(s)}$. First we examine the case (i) where $s(a) = \theta$ and $f^T(s)(a) = 1$. This means that for each $(b,a) \in R$ we have $s(b) = 0$ (we clearly have $b \in D_s$, as otherwise we do not get $f^T(s)(a) = 1$), but by Theorem 20 we get that $b \in S^+$ which means a is defended by S but not in S, contradicting completeness. We now examine the case (ii) where $s(a) = \theta$ and $f^T(s)(a) = 0$. This means there is some $(b,a) \in R$ with $f^T(s)(b) = 1$, and by case (i) this is not possible. Hence, we indeed get $D_s = D_{f^T(s)}$.

$\Leftarrow$: Assume s is a symbolic steady state. By Theorem 20 we get that S is admissible in T. Towards contradiction assume S is not complete, i.e., $a \notin S$ is defended by S (which means $s(a) = \theta$). This means for each $(b,a) \in R$ we have that $s(b) = 0$, which means $f^T(s)(b) = 1$, a contradiction, since we assumed s is a symbolic steady state. Hence, S is complete. □

6. Conclusions

In this work we established a connection between abstract argumentation frameworks and boolean networks. That is, we provided a one-to-one mapping between Dung style argumentation frameworks and the class boolean networks where transitions functions are conjunctions of negative literals, such that the AF coincidences with interaction graph of the BN. We have shown that stable extensions correspond to single state attractors, complete extensions correspond to symbolic steady states—in particular, the complete state pairs, and admissible sets correspond to seeds in the corresponding boolean network. This, on the one hand side allows us to use the formal machinery of boolean networks to reason on argumentation frameworks and on the other hand side argumentation semantics provide alternative characterizations in the "static" setting of the interaction graph of a BN for concepts that are rooted in the dynamic setting of the transition graph.

Given that in general boolean networks the transition function can be an arbitrary boolean function an interesting direction for future work is to investigate how richer abstract argumentation formalisms like Argumentation Frameworks with collective attacks (SETAFs) [13], Bipolar Argumentation [14], or even Abstract Dialectical Frameworks (ADFs) [15] relate to boolean networks, and vice versa how frequently studied classes of BNs [16,17,18], that put restrictions on the transition functions, relate to abstract argumentation.

Acknowledgments. This research has been supported by the Vienna Science and Technology Fund (WWTF) through project ICT19-065, and the Austrian Science Fund (FWF) through project P32830.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artif Intell. 1995;77(2):321-58.

[2] Akutsu T. Algorithms for Analysis, Inference, and Control of Boolean Networks. World Scientific; 2018.

[3] Kauffman S. Homeostasis and differentiation in random genetic control networks. Nature. 1969;224(5215):177-8.

[4] Thomas R. Boolean formalization of genetic control circuits. Journal of Theoretical Biology. 1973;42(3):563-85.

[5] Schwab JD, Kühlwein SD, Ikonomi N, Kühl M, Kestler HA. Concepts in Boolean network modeling: What do they all mean? Computational and Structural Biotechnology Journal. 2020;18:571-82.

[6] Mori T, Akutsu T. Attractor detection and enumeration algorithms for Boolean networks. Computational and Structural Biotechnology Journal. 2022;20:2512-20.

[7] Yang G, Albert R. Modeling of molecular networks. In: The Dynamics of Biological Systems. Springer International Publishing; 2019. p. 35-62.

[8] Thomas R, Kaufman M. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. Chaos. 2001;11(1):180 – 195.

[9] Siebert H. Analysis of discrete bioregulatory networks using symbolic steady states. Bulletin of mathematical biology. 2011;73:873-98.

[10] Klarner H, Bockmayr A, Siebert H. Computing symbolic steady states of boolean networks. In: Proc. of 11th International Conference on Cellular Automata for Research and Industry; 2014. p. 561-70.

[11] Richard A. Positive and negative cycles in Boolean networks. Journal of Theoretical Biology. 2019;463:67-76.

[12] Caminada M, Gabbay DM. A logical account of formal argumentation. Studia Logica. 2009;93(2):109-45.

[13] Bikakis A, Cohen A, Dvořák W, Flouris G, Parsons S. Joint Attacks and Accrual in Argumentation Frameworks. In: Gabbay D, Giacomin M, Simari GR, Thimm M, editors. Handbook of Formal Argumentation. vol. 2. College Publications; 2021. p. 131-210.

[14] Cayrol C, Cohen A, Lagasquie-Schiex MC. Higher-Order Interactions (Bipolar or not) in Abstract Argumentation: A State of the Art. In: Gabbay D, Giacomin M, Simari GR, Thimm M, editors. Handbook of Formal Argumentation. vol. 2. College Publications; 2021. p. 15-130.

[15] Brewka G, Ellmauthaler S, Strass H, Wallner JP, Woltran S. Abstract Dialectical Frameworks. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. vol. 1. College Publications; 2018. p. 237-85.

[16] Gao Z, Chen X, Başar T. Stability structures of conjunctive Boolean networks. Automatica. 2018;89:8-20.

[17] Veliz-Cuba A, Buschur K, Hamershock R, Kniss A, Wolff E, Laubenbacher R. AND-NOT logic framework for steady state analysis of Boolean network models. Appl Math Inf Sci. 2013;7(4):1263-74.

[18] Li Y, Adeyeye JO, Murrugarra D, Aguilar B, Laubenbacher R. Boolean nested canalizing functions: A comprehensive analysis. Theoretical Computer Science. 2013;481:24-36.