Graphical Analysis of Abstract Argumentation Frameworks via Boolean Networks

Van-Giang Trinh[1], Belaid Benhamou[2] and Vincent Risch[2]

[1]*Inria Saclay, EP Lifeware, Palaiseau, France*

[2]*LIRICA Team, LIS, Aix-Marseille University, Marseille, France*

van-giang.trinh@inria.fr,{belaid.benhamou, vincent.risch}@univ-amu.fr

Keywords: Abstract Argumentation Framework, Extension-Based Semantics, Graphical Analysis, Boolean Network.

Abstract: Abstract Argumentation Frameworks (AFs) are the key formalism of abstract argumentation, which is one of the main directions in argumentation research. An AF is mainly studied by means of its extensions, defined as subsets of arguments. In this work, we define a Boolean Network (BN) encoding for AFs, where BNs are a simple and efficient mathematical formalism that has a long history of research. We then show that the attack graph of an AF coincides with the influence graph of its encoded BN, and in particular preferred and stable extensions of this AF one-to-one correspond to minimal trap spaces and fixed points of the encoded BN, respectively. We also define a new concept for BNs called *complete* trap space, then show that complete trap spaces (resp. the percolation of the special trap space where all variables are free) in BNs one-to-one correspond (resp. corresponds) to complete extensions (resp. the grounded extension) in AFs. This connection opens the promising application to graphical analysis of AFs, which is an interesting line of research with many useful applications. More specifically, we use it to explore many new results relating extensions of an AF and (positive or negative) cycles in its attack graph. In particular, we show new upper bounds based on positive feedback vertex sets for the numbers of stable, preferred, and complete extensions.

1 INTRODUCTION

Abstract Argumentation Frameworks (AFs) are the key formalism of abstract argumentation, which is one of the main directions in argumentation research (Toulmin, 1958; Pollock, 1987; Pollock, 1991b; Pollock, 1991a; Dung, 1995; Baroni et al., 2020). An AF models arguments as vertices in a directed graph, where a directed arc denotes an *attack* from the starting vertex to the ending vertex, providing a graphical representation. The main concept to study AFs is an *extension* defined as a subset of arguments. There are many different types of extension-based semantics in AFs (Baroni et al., 2020). Among others, stable, preferred, grounded, and complete semantics are first proposed in Dung's 1995 seminal paper. Nowadays, they still play a central role in argumentation research and attract much attention from not only the argumentation community but also other research communities (Baumann and Strass, 2013; Thimm et al., 2021; Obiedkov and Sertkaya, 2023; Dimopoulos et al., 2024).

Regarding the analysis of AFs, there are two main directions of research. In practice, a various number of interesting computational problems w.r.t. extensions have been proposed and studied for decades (Charwat et al., 2015). Notable ones include 1) deciding a given argument appears in at least one extension (resp. all extensions) of a certain type, i.e., credulous (resp. skeptical) reasoning (Thimm et al., 2021), 2) enumerating extensions of a certain type (Kröll et al., 2017), and 3) counting the number of all extensions of a certain type, which is also a direct consequence of the enumeration problem (Dewoprabowo et al., 2022). To address these problems, many methods have been proposed, exploiting prominent techniques in symbolic AI such as answer set programming, SAT, and constraint programming, or techniques from other fields such as graph theory and formal concept analysis.

In theory, it is interesting and crucial to find graphical conditions for properties on extensions of an AF. For example, several studies (Baumann and Strass, 2013; Baumann and Strass, 2015; Ulbricht, 2021) investigated a basic question, namely how many extensions can an AF possess under a given semantics. The results of this line of research have many useful applications to abstract argumentation, for example they

Trinh, V.-G., Benhamou, B. and Risch, V.
Graphical Analysis of Abstract Argumentation Frameworks via Boolean Networks.
DOI: 10.5220/0013346400003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 2*, pages 745-756
ISBN: 978-989-758-737-5; ISSN: 2184-433X

can be used to provide lower bounds for the minimal realizability of certain sets of extensions (Baumann et al., 2014) and upper bounds for extension computation algorithms (Baumann and Strass, 2015). There are also other studies focusing on relationships among different semantics of an AF under certain graphical conditions (Dung, 1995; Yun et al., 2017). In this work, we focus on the graphical analysis of AFs.

Boolean Networks (BNs) are a simple and efficient mathematical formalism that has a long history of research and has been widely applied to many areas from science to engineering such as mathematics, computer science, neural networks, manufacturing, IoT, and in particular systems biology (Schwab et al., 2020). A BN is a discrete dynamical system including n Boolean variables associated with n Boolean functions to express the state update over discrete time following an employed update scheme. Recently, trap spaces have been proposed (Klarner et al., 2015), and now have become the central focus in the analysis and control of BNs (Rozum et al., 2021; Trinh et al., 2023; Trinh et al., 2024a; Trinh et al., 2024c). A trap space is a well-structured part of the state space where the BN's dynamics cannot escape once entered. If a trap space only contains one state, it is a fixed point. In contrast to other dynamical concepts in BNs, trap spaces (also fixed points) are independent of the employed update scheme. Very recently, BNs have been connected to logic programming, and then used to study the graphical analysis of normal logic programs (Trinh and Benhamou, 2024; Trinh et al., 2024b), which are closely related to AFs (Dung, 1995).

Motivated by the aforementioned elements, in this work, we establish a connection between AFs and BNs. More specifically, we define a BN encoding for AFs. We then show that the attack graph of an AF coincides with the influence graph of its encoded BN, and in particular preferred and stable extensions of this AF one-to-one correspond to minimal trap spaces and fixed points of the encoded BN, respectively. We also define a new concept for BNs called *complete trap space* (inspired by the concept of complete extension in AFs), then show that complete trap spaces (resp. the percolation of the special trap space where all variables are free) in BNs one-to-one correspond (resp. corresponds) to complete extensions (resp. the grounded extension) in AFs. This connection opens the promising application to the graphical analysis of AFs. We use it to explore many new results relating extensions of an AF and (positive or negative) cycles in its attack graph. In particular, we show new upper bounds based on positive feedback vertex sets for the numbers of stable, preferred, and complete extensions in AFs. Some of these results are quite straightforward consequences of existing graphical analysis results in the BN theory, but there are some results that rely on *new* results in the BN theory (including Theorem 12, Theorem 17, and Theorem 20) that we claim and formally prove.

In the preparation of the present manuscript, we have recently noticed that independently from us, three other groups of researchers have discovered the connection between AFs and BNs (Dimopoulos et al., 2024; Heyninck et al., 2024; Azpeitia et al., 2024). Although sharing some parts of results, our work contains many new results that do not exist in the others. We list here several notable ones:

- The bijection between the set of complete extensions of an AF and the set of complete trap spaces of its encoded BN (Theorem 1).
- The equivalence between the grounded extension of an AF and the percolation of the special trap space of its encoded BN (Theorem 2).
- A more general characterization of complete trap spaces in BNs (Theorem 4).
- Importantly, all the graphical analysis results shown in Section 5.

2 PRELIMINARIES

We use $\mathbb{B} = \{0,1\}$ as the Boolean domain and the logical connectives used in this paper are $\wedge$ (conjunction), $\vee$ (disjunction), and $\neg$ (negation).

2.1 Abstract Argumentation Frameworks

An Abstract Argumentation Framework (AF) is a tuple $\mathcal{A} = (A,R)$, where A is a finite set of arguments[1] and R is a binary *attack* relation on A. An AF $\mathcal{A}$ can be represented as a signed directed graph (called the *attack graph*) $\text{ag}(\mathcal{A}) = (V,E)$ where $V = A$ and $E = \{(ab, \ominus) \mid (a,b) \in R\}$ ($\ominus$ stands for the attack). Then a^- (resp. a^+) denotes the set of predecessors (resp. successors) of argument a in $\text{ag}(\mathcal{A})$, i.e., the set of arguments that attack a (resp. attacked by a). These two concepts can be extended for a subset S of arguments, i.e., $S^- = \bigcup_{a \in S} a^-$ and $S^+ = \bigcup_{a \in S} a^+$. Conventionally, $\emptyset^- = \emptyset^+ = \emptyset$. Given a subset $S \subseteq A$ (also called an *extension*). S is *conflict-free* iff there are no arguments a and b in S such that a attacks b, i.e., $(a,b) \in R$. An argument $a \in A$ is *acceptable* w.r.t.

[1]The abstract argumentation community mostly focuses on *finite* (instead of infinite) AFs.

S iff $\forall b \in A$: if b attacks a, then b is attacked by some argument in S. S is *admissible* iff it is conflict-free and each argument in S is acceptable w.r.t. S. S is a *stable extension* iff S is conflict-free and it attacks every argument that is not in S, i.e., $S^+ = A \setminus S$. S is a *preferred extension* iff it is a subset-maximal admissible set. S is a *complete extension* iff it is an admissible set such that for each a acceptable w.r.t. S, $a \in S$. S is a *grounded extension* iff it is a subset-minimal complete extension.

A related, and often interchangeable concept for extension is *labelling* introduced by (Caminada and Gabbay, 2009). A labelling is a mapping $\lambda\colon A \to \{\text{in}, \text{out}, \text{undec}\}$. The corresponding labelling of an extension S is $\gamma(S) = \{(a,\text{in}) \mid a \in S\} \cup \{(a,\text{out}) \mid a \in S^+\} \cup \{(a,\text{undec}) \mid a \in A \setminus (S \cup S^+)\}$. We define $\text{in}(\lambda) = \{a \in A \mid \lambda(a) = \text{in}\}$, $\text{out}(\lambda) = \{a \in A \mid \lambda(a) = \text{out}\}$, and $\text{undec}(\lambda) = \{a \in A \mid \lambda(a) = \text{undec}\}$. Then the corresponding extension of λ is $\text{in}(\lambda)$. From now on, we can use the terms of extension and labelling interchangeably. In addition, λ is a *complete labelling* iff for each $a \in A$, it holds that: if $\lambda(a) = \text{in}$ then $\lambda(b) = \text{out}$ for every $b \in a^-$; if $\lambda(a) = \text{out}$ then there exists $b \in a^-$ such that $\lambda(b) = \text{in}$; and if $\lambda(a) = \text{undec}$ then not every argument $b \in a^-$ has $\lambda(b) = \text{out}$ and there is no argument $b \in a^-$ has $\lambda(b) = \text{in}$. A complete labelling λ is *grounded* iff $\text{in}(\lambda)$ is subset-minimal. Given an AF, the grounded extension of this AF is always unique. A complete labelling λ is *preferred* iff $\text{in}(\lambda)$ is subset-maximal. Furthermore, if $\text{in}(\lambda)$ is subset-maximal (resp. subset-minimal), then $\text{out}(\lambda)$ is also subset-maximal (resp. subset-minimal). A preferred labelling λ is *stable* iff $\text{in}(\lambda) \cup \text{out}(\lambda) = A$. Complete, grounded, preferred, and stable labellings correspond to complete, grounded, preferred, and stable extensions, respectively (Caminada and Gabbay, 2009).

Example 1. *Let us consider AF $\mathcal{A}^1 = (A^1, R^1)$ with $A^1 = \{a,b,c\}$ and $R^1 = \{(a,b),(a,c),(b,a),(c,b),(c,c)\}$. The attack graph of $\mathcal{A}^1$ is given in Figure 1. $\mathcal{A}^1$ has three conflict-free sets: $S_1 = \emptyset$, $S_2 = \{a\}$, and $S_3 = \{b\}$. We have three corresponding labellings: $\gamma(S_1) = \{(a,\mathit{undec}), (b,\mathit{undec}), (c,\mathit{undec})\}$, $\gamma(S_2) = \{(a,\mathit{in}), (b,\mathit{out}), (c,\mathit{out})\}$, and $\gamma(S_3) = \{(a,\mathit{out}), (b,\mathit{in}), (c,\mathit{undec})\}$. S_1 and S_2 are admissible sets, which are also complete extensions of $\mathcal{A}^1$. S_1 is the unique grounded extension of $\mathcal{A}^1$. S_2 is a preferred (also stable) extension of $\mathcal{A}^1$. We also have that $\gamma(S_1)$ and $\gamma(S_2)$ are complete labellings, $\gamma(S_1)$ is the unique grounded labelling, and $\gamma(S_2)$ is a preferred (also stable) labelling of $\mathcal{A}^1$.*

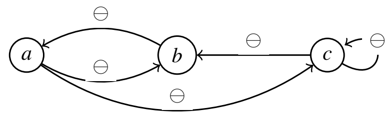


Figure 1: Attack graph of $\mathcal{A}^1$ shown in Example 1 and influence graph of f^1 shown in Example 2.

2.2 Boolean Networks

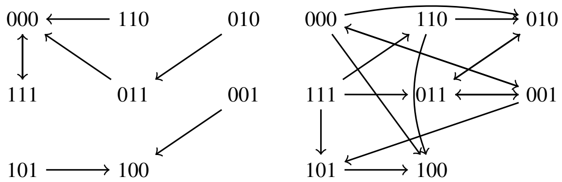
A Boolean Network (BN) f is a finite set of Boolean functions on a set of Boolean variables denoted by var_f. Each variable v is associated with a Boolean function $f_v\colon \mathbb{B}^{|\text{var}_f|} \to \mathbb{B}$. f_v is called *constant* if it is always either 0 or 1 regardless of the values of its arguments. A state s of f is a Boolean vector $s \in \mathbb{B}^{|\text{var}_f|}$. s can be seen as a mapping $s\colon \text{var}_f \to \mathbb{B}$. We write s_v to denote the value of variable v in s. For convenience, we write a state simply as a string of values of variables in this state (e.g., 0110 instead of (0, 1, 1, 0)).

Let x be a state of f. We use $x[v \leftarrow a]$ to denote the state y so that $y_v = a$ and $y_u = x_u, \forall u \in \text{var}_f, u \neq v$ where $a \in \mathbb{B}$. The Influence Graph (IG) of f (denoted by $\text{ig}(f)$) is a signed directed graph (V,E) on the set of signs $\{\oplus, \ominus\}$ where $V = \text{var}_f$, $(uv, \oplus) \in E$ (i.e., u positively affects the value of f_v) iff there is a state x such that $f_v(x[u \leftarrow 0]) < f_v(x[u \leftarrow 1])$, and $(uv, \ominus) \in E$ (i.e., u negatively affects the value of f_v) iff there is a state x such that $f_v(x[u \leftarrow 0]) > f_v(x[u \leftarrow 1])$. Let v^- (resp. v^+) denote the set of predecessors (resp. successors) of v in $\text{ig}(f)$. Then $|v^-|$ (resp. $|v^+|$) is called the in-degree (resp. out-degree) of v. The minimum in-degree of $\text{ig}(f)$ is defined as the smallest in-degree of all vertices v in $\text{ig}(f)$. Clearly, f contains no constant function iff the minimum in-degree of $\text{ig}(f)$ is at least one. A cycle (possibly a self loop) of a signed directed graph is positive (resp. negative) if its number of arcs is even (resp. odd). A positive (resp. negative) feedback vertex set is a set of vertices that intersect all positive (resp. negative) cycles.

At each time step t, variable v can update its state to $s'_v = f_v(s)$, where s (resp. s') is the state of f at time t (resp. $t+1$). An *update scheme* of a BN refers to how variables update their states over (discrete) time (Schwab et al., 2020). Various update schemes exist, but the primary types are *synchronous*, where all variables update simultaneously, and *fully asynchronous*, where a single variable is non-deterministically chosen for updating. By adhering to the employed update scheme, the BN transitions from one state to another, which may or may not be the same. This transition is referred to as the *state transition*. Then the dynamics of the BN is captured by a directed graph referred to as the *State Transition Graph* (STG). We use $\text{sstg}(f)$ (resp. $\text{astg}(f)$) to denote the STG of f under the synchronous (resp. fully asynchronous) update scheme.

A non-empty set of states is a *trap set* if it has no out-going arcs on the STG of f. An *attractor* is a subset-minimal trap set. An attractor of size 1 (resp. at least 2) is called a fixed point (resp. cyclic attractor). A *sub-space* m of a BN f is a mapping $m\colon \text{var}_f \to \mathbb{B}_\star$ where $\mathbb{B}_\star = \mathbb{B} \cup \{\star\}$ denoting the three-valued domain. A variable $v \in \text{var}_f$ is called *fixed* (resp. *free*) in m iff $m(v) \neq \star$ (resp. $m(v) = \star$). A sub-space m represents a set of states denoted by $\mathcal{S}[m]$ such that $\mathcal{S}[m] = \{s \in \mathbb{B}^{|\text{var}_f|} | s_v = m(v), \forall v \in \text{var}_f, m(v) \neq \star\}$. For example, $m = \{v_1 = \star, v_2 = 1, v_3 = 1\}$ and $\mathcal{S}[m] = \{011, 111\}$. If a sub-space is also a trap set, it is a *trap space*. Unlike trap sets and attractors, trap spaces of a BN are independent of the employed update scheme (Klarner et al., 2015). In particular, a fixed point of f is a special trap space where no variable is mapped to $\star$. A trap space m is minimal iff there is no trap space m' such that $\mathcal{S}[m'] \subset \mathcal{S}[m]$. Since an attractor is a subset-minimal trap set, a minimal trap space contains at least one attractor of the BN regardless of the employed update scheme.

Example 2. *Let us consider BN f^1 with var$_{f^1}$ = $\{a,b,c\}$, $f^1_a = \neg b$, $f^1_b = \neg a \wedge \neg c$, and $f^1_c = \neg a \wedge \neg c$. The IG of f^1 is given in Figure 1. Figures 2(a) and 2(b) respectively show the synchronous and asynchronous STGs of f^1 where self arcs are omitted for simplicity. sstg(f^1) has one fixed point ($\{100\}$) and one cyclic attractor ($\{000, 111\}$). astg(f^1) has only one fixed point ($\{100\}$). f^1 has three trap spaces (same in the both STGs): $m_1 = \{a = \star, b = \star, c = \star\}$, $m_2 = \{a = 1, b = 0, c = 0\}$, $m_3 = \{a = 1, b = 0, c = \star\}$. Then m_2 is a minimal trap space of f^1.*



(a) . (b) .

Figure 2: (a) sstg(f^1) and (b) astg(f^1). f^1 is given in Example 2.

3 RELATED WORK

3.1 Connections with Other Theories

AFs are closely connected to logic programming, one of non-monotonic reasoning frameworks, starting from the early studies (Pollock, 1991b; Pollock, 1991a; Dung, 1995). Subsequent studies of this direction (Caminada and Gabbay, 2009; Caminada et al., 2015) showed more clearly the equivalence between extensions in AFs and models in logic programs such as stable extensions vs. stable models, complete extensions vs. stable partial models, and preferred extensions vs. regular models. Key to prove the equivalence is the use of labellings (Caminada and Gabbay, 2009). There are also some studies trying to encode preferred extensions as stable models of logic programs (Nieves et al., 2008). Furthermore, AFs were also connected to default theories (Nouioua and Risch, 2012), thus it showed that any admissible (or preferred) set of arguments of an AF can be directly computed from the ι-answer sets of its equivalent logic program.

Because of the intuitive formalization using directed graphs, AFs were naturally connected to graph theory (Dimopoulos and Torres, 1996). Several equivalence results have been obtained, not only pointing out computational tractable classes of AFs under certain graph-theoretic constraints (Dunne, 2007) but also contributing to the analysis of AFs (Gaspers and Li, 2019). Recently, AFs have been connected to lattices (Elaroussi et al., 2023) (in terms of preferred extensions) and formal concept analysis (Obiedkov and Sertkaya, 2023) (in terms of stable extensions).

3.2 Graphical Analysis

In his 1995 seminal paper, Dung also provided some essential results regarding the graphical analysis of AFs (Dung, 1995). For example, he showed that an AF without cycles in its attack graph has exactly one complete extension that is also preferred and stable. He also showed that if the attack graph has no negative cycles, then the stable and preferred extensions of the AF coincide, leading to it has at least one stable extension. Subsequent studies (Baumann and Strass, 2013; Ulbricht, 2021; Baumann and Ulbricht, 2021) dived deeply into the question of how many extensions can an AF possess under a given semantics. The work by (Baumann and Strass, 2013) presented a first analytical and empirical study of the maximal and average numbers of stable extensions, in particular showing that for any AF of n arguments, the number of stable extensions is at most $3^{\frac{n}{3}}$. This number was latter shown to be an upper bound for the number of preferred extensions (Dunne et al., 2015). Recently, the work by (Ulbricht, 2021) has answered a reasonable conjecture claimed in (Baumann and Strass, 2015) on the maximal number of complete extensions. More specifically, it shows that the number of complete extensions of any AF of n arguments is at most $3^{\frac{n}{2}}$. Finally, a branch of this research direction is to investigate graphical properties of special types of AFs such as symmetric AFs (Coste-Marquis et al., 2005).

4 AF-BN CONNECTION

4.1 BN Encoding

We first define a BN encoding of AFs as follows.

Definition 1. *Let $\mathcal{A} = (A,R)$ be an AF. Its encoded BN f is: $var_f = A$, $f_a = \bigwedge_{b \in a^-} \neg b, \forall a \in A$. If $a^- = \emptyset$, then $f_a = 1$.*

A BN f is called *negative AND-NOT* iff every its update function is only a conjunction of negative literals (Richard and Ruet, 2013). Clearly, a negative AND-NOT BN is uniquely determined by its influence graph. In particular, the encoded BN of an AF is a negative AND-NOT BN.

A straightforward consequence from the encoding is:

Proposition 1. *Let $\mathcal{A} = (A,R)$ be an AF and f be its encoded BN. Then $ag(\mathcal{A}) = ig(f)$.*

Indeed, the example BN f^1 (see Example 2) is the encoded BN of the example AF $\mathcal{A}^1$ (see Example 1). The influence graph of f^1 and the attack graph of $\mathcal{A}^1$ are the same (see Figure 1). $\text{ig}(f^1)$ has one positive cycle ($a \xrightarrow{\ominus} b \xrightarrow{\ominus} a$) and two negative cycles ($c \xrightarrow{\ominus} c$ and $a \xrightarrow{\ominus} c \xrightarrow{\ominus} b \xrightarrow{\ominus} a$). By considering in as 1, out as 0, and undec as $\star$, we can obtain the equivalence between labellings in an AF and sub-spaces in a BN. From now on, we can use these terms interchangeably. As such, trap space m_1 (resp. m_2) of f^1 is equivalent to complete labelling $\gamma(S_1)$ (resp. $\gamma(S_2)$) of $\mathcal{A}^1$. Trap space m_3 of f^1 does not correspond to any complete labelling of $\mathcal{A}^1$. However, we can see the equivalence between minimal trap spaces of f^1 (i.e., m_2) and preferred labllings of $\mathcal{A}^1$ (i.e., $\gamma(S_2)$). In addition, m_1 is equivalent to the grounded labelling $\gamma(S_1)$.

4.2 Complete Extensions

First, we define the order $\leq_s$ on $\mathbb{B}_\star$ by $0 <_s \star$, $1 <_s \star$, and $\leq_s$ contains no other relation. Then for two sub-spaces m_1 and m_2, we have $m_1 \leq_s m_2$ iff $m_1(a) \leq_s m_2(a), \forall a \in \text{var}_f$. It is also similar for labellings. In addition, $m_1 \leq_s m_2$ iff $\mathcal{S}[m_1] \subseteq \mathcal{S}[m_2]$, and $m_1 <_s m_2$ iff $m_1 \leq_s m_2$ and $m_1 \neq m_2$. It follows that $\leq_s$-minimal trap spaces are exactly minimal trap spaces and preferred labellings are exactly $\leq_s$-minimal complete labellings. Second, we define the truth order $\leq_t$ on $\mathbb{B}_\star$ by $0 <_t \star <_t 1$. Let e be a propositional formula on var_f. Then the evaluation of e under a sub-space m (denoted by $m(e)$) is defined recursively as follows:

$$m(e) = \begin{cases} m(a) & \text{if } e = a, a \in \text{var}_f \\ \neg m(e_1) & \text{if } e = \neg e_1 \\ \min_{\leq_t}(m(e_1), m(e_2)) & \text{if } e = e_1 \wedge e_2 \\ \max_{\leq_t}(m(e_1), m(e_2)) & \text{if } e = e_1 \vee e_2 \end{cases}$$

where $\neg 1 = 0, \neg 0 = 1, \neg\star = \star$, and $\min_{\leq_t}$ (resp. $\max_{\leq_t}$) is the function to get the minimum (resp. maximum) value of two values w.r.t. the order $\leq_t$. Note that if s is a state (i.e., a special sub-space where no variable is mapped to $\star$), then $s(e) = e(s)$ as s is also a vector of Boolean values. We have the following property for trap spaces.

Definition 2. *Given a BN f. We define $T(f)$ as the set of all sub-spaces m such that $m(f_a) \leq_s m(a)$ for every $a \in var_f$.*

Proposition 2. *Let f be a BN. Then $T(f)$ is exactly the set of all trap spaces of f.*

Proof. Let m be a sub-space of f. Let s be a state in $\mathcal{S}[m]$. It follows that $s \leq_s m$. Let s' be a successor state of s following the employed update scheme of f. For any variable $a \in \text{var}_f$, if a is updated, then $s'_a = f_a(s) = s(f_a)$, and $s'_a = s_a$ otherwise. We also have that $s(f_a) \leq_s m(f_a), \forall a \in \text{var}_f$ because $s \leq_s m$. Then $m \in T(f)$ iff $m(f_a) \leq_s m(a)$ for every $a \in \text{var}_f$ by definition iff $s'_a \leq_s m(a)$ for every $a \in \text{var}_f$ iff $s' \in \mathcal{S}[m]$ (regardless of the employed update scheme) iff m is a trap space of f. □

By Proposition 2, we can use $T(f)$ as the set of all trap spaces of f. Next, inspired by the concept of complete extension in AFs, we propose a new concept called *complete trap space* for BNs (see Definition 3).

Definition 3. *Given a BN f. A sub-space m is called a* comple trap space *iff $m(f_a) = m(a)$ for every $a \in var_f$. We denote $\widehat{T}(f)$ be the set of all complete trap spaces of f.*

Considering the example BN f^1 (given in Example 2), we have $T(f^1) = \{m_1, m_2, m_3\}$, whereas $\widehat{T}(f^1) = \{m_1, m_2\}$. Of course, we can see that $\widehat{T}(f) \subseteq T(f)$ for any BN f by definition. We then prove a deeper relationship between trap spaces in $\widehat{T}(f)$ and trap spaces in $T(f)$.

Proposition 3. *Let f be a BN. For every $m \in T(f)$, there is a trap space $\widehat{m} \in \widehat{T}(f)$ such that $\widehat{m} \leq_s m$.*

Proof. Let m^j be an arbitrary trap space in $T(f)$. We construct a sub-space m^{j+1} as $m^{j+1}(a) = m^j(f_a), \forall a \in \text{var}_f$.

We prove that m^{j+1} is also a trap space of f. By construction, it is a sub-space. In addition, $m^{j+1}(a) \leq_s m^j(a), \forall a \in \text{var}_f$ because m^j is a trap space, thus $m^{j+1} \leq_s m^j$. Let s be an arbitrary state in $\mathcal{S}[m^{j+1}]$. Of course, it is also in $\mathcal{S}[m^j]$ because $m^{j+1} \leq_s m^j$. Let s' be the next state of s on $\text{sstg}(f)$ (the synchronous STG of f), i.e., $s'_a = s(f_a), \forall a \in \text{var}_f$. Consider variable $a \in \text{var}_f$. Since $s \in \mathcal{S}[m^j]$, we have that $s(f_a) \leq_s m^j(f_a)$, leading to $s'_a \leq_s m^{j+1}(a)$. Hence, $s' \in \mathcal{S}[m^{j+1}]$, i.e., $\mathcal{S}[m^{j+1}]$ is a trap set of $\text{sstg}(f)$. It follows that m^{j+1} is a trap space.

Assume that m is a trap space in $T(f)$. We start with $m^j = m$ and repeat the above process by increasing j, and finally reach the case $m^{j+1} = m^j$ because $\mathcal{S}[m]$ is finite. By construction, $m^j(a) = m^j(f_a), \forall a \in \text{var}_f$ (thus $m^j \in \widehat{T}(f)$) and $m^j \leq_s m$. By setting $\widehat{m} = m^j$, we can conclude the proof. □

We then show that $\widehat{T}(f)$ is exactly the set of all complete labellings of the AF $\mathcal{A}$.

Theorem 1. *Let $\mathcal{A} = (A, R)$ be an AF and f be its encoded BN. The set of complete labellings of $\mathcal{A}$ coincides with the set $\widehat{T}(f)$.*

Proof. By construction, $\text{var}_f = A$. Let λ be a labelling of $\mathcal{A}$. It can be considered as a sub-space of f. Consider an argument $a \in A$. Recall that $f_a = \bigwedge_{b \in a^-} \neg b$. We have by definition that $\lambda(b) = \text{out}$ for every $b \in a^-$ iff $\lambda(f_a) = 1$. There is $b \in a^-$ such that $\lambda(b) = \text{in}$ iff $\lambda(f_a) = 0$. Not every argument $b \in a^-$ has $\lambda(b) = \text{out}$ and there is no argument $b \in a^-$ has $\lambda(b) = \text{in}$ iff $\lambda(f_a)$ can be neither 0 nor 1 iff $\lambda(f_a) = \star$. This implies that λ is a complete labelling of $\mathcal{A}$ iff $\lambda(a) = \lambda(f_a), \forall a \in A$ iff λ is a trap space in $\widehat{T}(f)$. □

4.3 Grounded Extensions

We here show that the grounded extension of an AF corresponds to a special complete trap space of its encoded BN (Theorem 2).

Definition 4 ((Trinh et al., 2024c)). *Given a sub-space m, the single-step percolation operator $\mathcal{P}$ produces a sub-space (denoted by $\mathcal{P}(m)$) with fixed variables given by those of m together with the free variables of m whose update functions are invariant on m. Formally, $\mathcal{P}(m)(v) = m(v)$ if $m(v) \neq \star$, $\mathcal{P}(m)(v) = b \in \mathbb{B}$ if $m(v) = \star$ and $f_v(x) = b, \forall x \in \mathcal{S}[m]$, and $\mathcal{P}(m)(v) = \star$ otherwise. The percolation operator $\mathcal{P}^\omega$ is obtained by repeated application of the single-step percolation operator $\mathcal{P}$ until a fixpoint, which always exists and can be achieved after up to $|var_f|$ application times because var_f is finite. We call $\mathcal{P}^\omega(m)$ the* percolation *of m.*

Proposition 4. *Given a BN f. If m is a trap space of f, then $\mathcal{P}^\omega(m)$ is a complete trap space of f.*

Proof. It is easy to see that if m is a trap space, then $\mathcal{P}(m)$ is unique and also a trap space. Hence, $\mathcal{P}^\omega(m)$ is unique and also a trap space in which Boolean functions of free variables cannot be simplified further under $\mathcal{P}^\omega(m)$. Of course, $\mathcal{P}^\omega(m)(f_v) = \mathcal{P}^\omega(m)(v)$ for every fixed variable v in $\mathcal{P}^\omega(m)$. For every free variable v in $\mathcal{P}^\omega(m)$, $\mathcal{P}^\omega(m)(f_v) = \star = \mathcal{P}^\omega(m)(v)$. By definition, $\mathcal{P}^\omega(m)$ is a complete trap space. □

Theorem 2. *Let $\mathcal{A}$ be an AF and f be its encoded BN. The grounded labelling of $\mathcal{A}$ equals to the percolation of sub-space ε where $\varepsilon(v) = \star, \forall v \in var_f$.*

Proof. Of course, ε is a trap space of f. By Proposition 4, $\mathcal{P}^\omega(\varepsilon)$ is a complete trap space of f.

For any trap spaces m_1 and m_2 of f, $\mathcal{P}(m_1) \leq_s \mathcal{P}(m_2)$ if $m_1 \leq_s m_2$. It follows that $\mathcal{P}^\omega(m_1) \leq_s \mathcal{P}^\omega(m_2)$ if $m_1 \leq_s m_2$. Since $m \leq_s \varepsilon$ for every trap space m, $\mathcal{P}^\omega(m) \leq_s \mathcal{P}^\omega(\varepsilon)$. Since $\mathcal{P}^\omega(m)$ is a complete trap space, $\mathcal{P}^\omega(\varepsilon)$ is the (unique) $\leq_s$-maximal complete trap space of f.

By definition, the grounded labelling of $\mathcal{A}$ is the unique subset-minimal complete labelling of $\mathcal{A}$. We have that $\leq_s$-maximal equals to subset-minimal. By Theorem 1, the grounded labelling of $\mathcal{A}$ equals to $\mathcal{P}^\omega(\varepsilon)$. □

4.4 Preferred and Stable Extensions

First, we show that preferred extensions of an AF $\mathcal{A}$ one-to-one correspond to minimal trap spaces of its encoded BN f.

Theorem 3. *Let $\mathcal{A} = (A, R)$ be an AF and f be its encoded BN. The set of preferred labellings of $\mathcal{A}$ coincides with the set of minimal trap spaces of f.*

Proof. We first show that a sub-space m is a $\leq_s$-minimal trap space w.r.t. $T(f)$ iff it is a $\leq_s$-minimal trap space w.r.t. $\widehat{T}(f)$. The "⇒" direction is trivial, since $\widehat{T}(f) \subseteq T(f)$. For the "⇐" direction, assume that m is not $\leq_s$-minimal w.r.t. $T(f)$. Then there is a trap space m' in $T(f)$ such that $m' <_s m$. By Proposition 3, there is a trap space $\widehat{m} \in \widehat{T}(f)$ such that $\widehat{m} \leq_s m'$. Then $\widehat{m} <_s m$, which is a contradiction because m is $\leq_s$-minimal w.r.t. $\widehat{T}(f)$. Hence, m is also a $\leq_s$-minimal trap space w.r.t. $T(f)$.

Now, we have that the set of minimal trap spaces of f coincides with the set of $\leq_s$-minimal trap spaces of $\widehat{T}(f)$. By Theorem 1, the set of complete labellings of $\mathcal{A}$ coincides with $\widehat{T}(f)$. Recall that a preferred labelling is a $\leq_s$-minimal complete labelling. Hence, the set of preferred labellings of $\mathcal{A}$ coincides with the set of minimal trap spaces of f. $\square$

Actually, Theorem 3 has been also claimed and proved in (Heyninck et al., 2024; Azpeitia et al., 2024) (but not in (Dimopoulos et al., 2024)) using a different way of proof. Next, we show the following corollary. This implies that stable extensions of $\mathcal{A}$ one-to-one correspond to fixed points of f.

Corollary 1. *Let $\mathcal{A} = (A,R)$ be an AF and f be its encoded BN. The set of stable labellings of $\mathcal{A}$ coincides with the set of fixed points of f.*

Proof. λ is a stable labelling of $\mathcal{A}$ iff it is a preferred labelling and $\lambda(a) \neq \text{undec}, \forall a \in A$ iff λ is a minimal trap space of f by Theorem 3 and $\lambda(a) \neq \star, \forall a \in \text{var}_f$ iff λ is a fixed point of f. $\square$

4.5 Characterization of Complete Trap Spaces

To end this section, we show a more general characterization of complete trap spaces in BNs (Theorem 4). Let $[S]$ denote the smallest sub-space that induces a set of states S (i.e., $S \subseteq \mathcal{S}[[S]]$). Let $\text{next}(S)$ denote the set of next states in the synchronous STG of states in S.

Theorem 4. *Given a BN f and a sub-space m of f. m is a complete trap space of f iff $m = [next(\mathcal{S}[m])]$.*

Proof. "$\Rightarrow$" Assume that m is a complete trap space. For every $v \in \text{var}_f$ such that $m(v) \neq \star$, we have $s_v = m(v), \forall s \in \text{next}(\mathcal{S}[m])$. For every $v \in \text{var}_f$ such that $m(v) = \star$, f_v cannot be simplified further under m, thus there always exists state s (resp. s') in $\mathcal{S}[m]$ such that $f_v(s) = 0$ (resp. $f_v(s') = 1$). It follows that $[\text{next}(\mathcal{S}[m])](v) = \star$. Hence, $m = [\text{next}(\mathcal{S}[m])]$.

"$\Leftarrow$" Assume that $m = [\text{next}(\mathcal{S}[m])]$. Then $\text{next}(\mathcal{S}[m]) \subseteq \mathcal{S}[m]$ by definition, implying that $\mathcal{S}[m]$ is a trap set in $\text{sstg}(f)$, thus m is a trap space of f. Suppose that m is not complete. It follows that $\mathcal{P}^{\omega}(m) <_s m$ (see the proof of Theorem 2), since $\mathcal{P}^{\omega}(m)$ is a complete trap space by Proposition 4. Then there is a variable $v \in \text{var}_f$ such that $m(v) = \star$, but $\mathcal{P}(m)(v) \neq \star$. This implies that $\forall s \in \text{next}(\mathcal{S}[m])$, $s_v = \mathcal{P}(m)(v)$, thus $m \neq [\text{next}(\mathcal{S}[m])]$, which is a contradiction. Hence, m is a complete trap space. $\square$

The general characterization shown in Theorem 4 relies on next states in the synchronous STG. It can be applicable for any BN. In contrast, the characterization shown in (Dimopoulos et al., 2024) relies on two-state attractors in the the synchronous STG and is applicable for only negative AND-NOT BNs.

Example 3. *Consider the BN f^1 given in Example 2. f^1 is the encoded BN of the AF $\mathcal{A}^1$ given in Example 1. f^1 has three trap spaces: $m_1 = \{a = \star, b = \star, c = \star\}$, $m_2 = \{a = 1, b = 0, c = 0\}$, $m_3 = \{a = 1, b = 0, c = \star\}$. m_1 and m_2 are complete. We have $[next(\mathcal{S}[m_1])] = [\{000, 011, 100, 111\}] = \{a = \star, b = \star, c = \star\} = m_1$ and $[next(\mathcal{S}[m_2])] = [\{100\}] = \{a = 1, b = 0, c = 0\} = m_2$. m_3 is not complete and $[next(\mathcal{S}[m_3])] = [\{100\}] = \{a = 1, b = 0, c = 0\} \neq m_3$.*

5 GRAPHICAL ANALYSIS RESULTS

The graphical analysis of BNs has a long history of research since BNs were originated (Kauffman, 1969). Nowadays, this line of research is still active with many prominent and deep results obtained, for example, the relationships between fixed points or attractors and positive or negative cycles in the influence graph, the upper bounds for numbers of fixed points or attractors based on feedback vertex sets in the influence graph. See (Paulevé and Richard, 2012; Richard, 2019) for more detailed reviews. The established connection between AFs and BNs opens the door to exploit these results for the graphical analysis of AFs, an interesting and crucial line of research on abstract argumentation. An advantage of the above approach is that we now can focus only on the dynamical properties in BNs when studying extensions in AFs.

5.1 Negative Cycles

First, we provide new proofs for the two known results presented in (Dung, 1995).

Theorem 5 (Theorem 33(1) of (Dung, 1995))**.** *Given an AF $\mathcal{A}$. If $ag(\mathcal{A})$ has no negative cycle, then all preferred extensions of $\mathcal{A}$ are stable.*

New proof. Let f be the encoded BN of $\mathcal{A}$. By Proposition 1, $\text{ig}(f)$ has no negative cycle. By Theorem 1 of (Richard, 2010), $\text{astg}(f)$ has no cyclic attractor. Each minimal trap space of f contains at least one attractor of $\text{astg}(f)$ (Klarner et al., 2015). In addition, if a minimal trap space contains a fixed point, then it is also a fixed point because of the minimality. Hence, all minimal trap spaces of f are fixed points. By Theorem 3 and Corollary 1, we can conclude that all preferred extensions of $\mathcal{A}$ are stable. $\square$

Corollary 2. *Given an AF $\mathcal{A}$. If $ag(\mathcal{A})$ has no negative cycle, then $\mathcal{A}$ has at least one stable extension.*

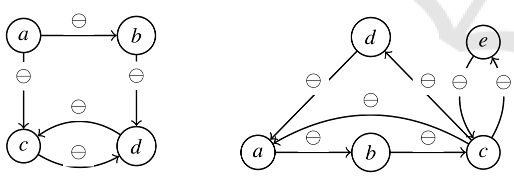
Proof. By Theorem 5, all preferred extensions of $\mathcal{A}$ are stable. Since $\mathcal{A}$ has at least one preferred extension (Dung, 1995), it has at least one stable extension. □

Next, we show a new result on stable extensions (Theorem 7).

Definition 5 ((Richard and Ruet, 2013)). *Given a signed directed graph G without positive arcs and vertices u, v of G (not necessarily distinct). A vertex $w \neq u, v$ is said to be a* subdivision *of (u,v) when 1) $(uw, \ominus)$ and $(wv, \ominus)$ are arcs of G; 2) $(uv, \ominus)$ is not an arc of G; 3) the in-degree and out-degree of w both equal 1.*

Definition 6 ((Richard and Ruet, 2013)). *Given a signed directed graph G without positive arcs, a cycle C of G and vertices u, v_1, v_2 of G. (u, v_1, v_2) is called a* killing triple *of C when 1) v_1 and v_2 are distinct vertices of C; 2) (u, v_1) has a subdivision in G, but no subdivision of (u, v_1) belongs to C; 3) $(uv_2, \ominus)$ is an arc of G that is not in C. A killing triple (u, v_1, v_2) of C is* internal *when u is a vertex in C,* external *otherwise.*

Example 4. *Consider two signed directed graphs without positive arcs: G_1 and G_2 (see Figures 3(a) and 3(b), respectively). In G_1, (a,d,c) is an external killing triple of positive cycle $c \xrightarrow{\ominus} d \xrightarrow{\ominus} c$ with b is a subdivision of (a,d). In G_2, (c,c,a) is an internal killing triple of positive cycle $a \xrightarrow{\ominus} b \xrightarrow{\ominus} c \xrightarrow{\ominus} d \xrightarrow{\ominus} a$ with e is a subdivision of (c,c). Positive cycle $c \xrightarrow{\ominus} e \xrightarrow{\ominus} c$ has no killing triple in G_2.*



(a) . (b) .

Figure 3: (a) G_1 and (b) G_2. These graphs are considered in Example 4.

Theorem 6 ((Richard and Ruet, 2013)). *Given a negative AND-NOT BN f. If every negative cycle of $ig(f)$ has an internal killing triple, then f has at least one fixed point.*

Theorem 7. *Given an AF $\mathcal{A}$. If every negative cycle of $ag(\mathcal{A})$ has an internal killing triple, then $\mathcal{A}$ has at least one stable extension.*

Proof. Let f be the encoded BN of $\mathcal{A}$. By Proposition 1, every negative cycle of $\text{ig}(f)$ has an internal killing triple. By Theorem 6, f has at least one fixed point. By Corollary 1, $\mathcal{A}$ has at least one stable extension. □

Indeed, $\text{ag}(\mathcal{A})$ has no negative cycle implies that every negative cycle of $\text{ag}(\mathcal{A})$ has an internal killing triple holds true. Hence, Theorem 7 is stronger than Corollary 2.

5.2 Positive Cycles

We first show that the presence of positive cycles in the attack graph is the necessary condition for the existence of multiple stable or preferred extensions.

Theorem 8 ((Aracena, 2008)). *Let f be a BN. If $ig(f)$ has no positive cycle, then f has at most one fixed point.*

Theorem 9. *Given an AF $\mathcal{A}$. If $ag(\mathcal{A})$ has no positive cycle, then $\mathcal{A}$ has at most one stable extension.*

Proof. Let f be the encoded BN of $\mathcal{A}$. By Proposition 1, $\text{ig}(f)$ has no positive cycle. By Theorem 8, f has at most one fixed point. By Corollary 1, $\mathcal{A}$ has at most one stable extension. □

Theorem 10 ((Richard and Comet, 2007)). *Let f be a BN. If $ig(f)$ has no positive cycle, then $astg(f)$ has a unique attractor.*

Theorem 11. *Given an AF $\mathcal{A}$. If $ag(\mathcal{A})$ has no positive cycle, then $\mathcal{A}$ has a unique preferred extension.*

Proof. Let f be the encoded BN of $\mathcal{A}$. By Proposition 1, $\text{ig}(f)$ has no positive cycle. By Theorem 10, $\text{astg}(f)$ has a unique attractor. Each minimal trap space of f contains at least one attractor of $\text{astg}(f)$ (Klarner et al., 2015) and $\mathcal{S}[m_1] \cap \mathcal{S}[m_2] = \emptyset$ for any two distinct minimal trap spaces m_1, m_2 (Trinh et al., 2023). It follows that f has at most one minimal trap space. Since f has at least one minimal trap space, f has a unique minimal trap space. By Theorem 3, $\mathcal{A}$ has a unique preferred extension. □

Surprisingly, this condition also holds true for complete extensions (Theorem 13). Note that the proof for this result relies on Theorem 12 that is new in the BN theory.

Lemma 1. *Let f be the negative AND-NOT BN. Assume that $ig(f)$ has the minimum in-degree of at least one. If $ig(f)$ has no positive cycle, then f has a unique complete trap space.*

Proof. Since $\text{ig}(f)$ has the minimum in-degree of at least one, f has no constant function. Then the subspace ε where all variables are free is simply a complete trap space of f.

Assume that f has a complete trap space $m \neq \varepsilon$. Then there exists a variable v_0 such that $m(v_0) \neq \star$. If $m(v_0) = 1$, then $m(f_{v_0}) = 1$ because m is complete, leading to $m(v) = 0$ for every $v \in v_0^-$ (as f_{v_0}

is a conjunction of negative literals), thus there is $v_1 \in v_0^-$ such that $m(v_1) = 0$ because $\text{ig}(f)$ has the minimum in-degree of at least one. If $m(v_0) = 0$, then $m(f_{v_0}) = 0$ because m is complete, leading to there is $v_1 \in v_0^-$ such that $m(v_1) = 1$. Repeating this reasoning, we have an infinite descending chain $v_0 \xleftarrow{\ominus} v_1 \xleftarrow{\ominus} v_2 \ldots$ such that $m(v_i) \neq \star, \forall i \geq 0$ and $m(v_{i+1}) = \neg m(v_i)$. Since var_f is finite, there exist two integer numbers j and k ($j, k \geq 0$) such that $v_j = v_{j+k}$ in the infinite descending chain, i.e., $C = v_j \xleftarrow{\ominus} v_{j+1} \xleftarrow{\ominus} \ldots \xleftarrow{\ominus} v_{j+k-1} \xleftarrow{\ominus} v_j$ is a cycle of $\text{ig}(f)$. We have that $m(v_j) = m(v_{j+2}) = \ldots$, thus k is even. Hence, C is a positive cycle, which is a contradiction.

Now we can conclude that f has a unique complete trap space. □

Theorem 12. *Given a negative AND-NOT BN f. If $\text{ig}(f)$ has no positive cycle, then f has a unique complete trap space.*

Proof. By percolating constant functions of f (similar to the percolation on trap spaces shown in Definition 4), we get either a non-empty BN f' without constant functions or an empty BN. In the latter case, we have that f has a unique complete trap space. In the former case, we have that f' is a negative AND-NOT BN and has no constant function, equivalently $\text{ig}(f')$ has the minimum in-degree of at least one. $\text{ig}(f)$ has no positive cycle, thus $\text{ig}(f')$ has no positive cycle because $\text{ig}(f')$ is clearly a sub-graph of $\text{ig}(f)$. By Lemma 1, f' has a unique complete trap space. Clearly, there is a bijection between the set of complete trap spaces of f and that of f'. Hence, f has a unique complete trap space. □

Theorem 13. *Given an AF $\mathcal{A}$. If $ag(\mathcal{A})$ has no positive cycle, then $\mathcal{A}$ has a unique complete extension.*

Proof. It straightforwardly follows from Proposition 1, Theorem 1, and Theorem 12. □

Finally, we show a stronger result of Theorem 9.

Theorem 14 ((Richard and Ruet, 2013))**.** *Given a negative AND-NOT BN f. If every positive cycle of $\text{ig}(f)$ has a killing triple, then f has at most one fixed point.*

Theorem 15. *Given an AF $\mathcal{A}$. If every positive cycle of $ag(\mathcal{A})$ has a killing triple, then $\mathcal{A}$ has at most one stable extension.*

Proof. Let f be the encoded BN of $\mathcal{A}$. By Proposition 1, every positive cycle of $\text{ig}(f)$ has a killing triple. By Theorem 14, f has at most one fixed point. By Corollary 1, $\mathcal{A}$ has at most one stable extension. □

5.3 Upper Bounds

Hereafter, we show three new upper bounds for the numbers of preferred, stable, and complete extensions of an AF, respectively. To the best of our knowledge, they are the first results relating the numbers of preferred, stable, and complete extensions with positive feedback vertex sets of the attack graph.

Theorem 16. *Given an AF $\mathcal{A}$. Let U be a subset of vertices that intersects every positive cycle of $ag(\mathcal{A})$. Then $\mathcal{A}$ has at most $2^{|U|}$ preferred extensions.*

Proof. Let f be the encoded BN of $\mathcal{A}$. By Proposition 1, U intersects every positive cycle of $\text{ig}(f)$. By Corollary 2 of (Richard, 2009), $\text{astg}(f)$ has at most $2^{|U|}$ attractors. It follows that f has at most $2^{|U|}$ minimal trap spaces. By Theorem 3, we can conclude that $2^{|U|}$ is an upper bound for the number of preferred extensions of $\mathcal{A}$. □

Corollary 3. *Given an AF $\mathcal{A}$. Let U be a subset of vertices that intersects every positive cycle of $ag(\mathcal{A})$. Then $\mathcal{A}$ has at most $2^{|U|}$ stable extensions.*

Proof. By Theorem 16, $\mathcal{A}$ has at most $2^{|U|}$ preferred extensions. A stable extension is also a preferred extension, thus $\mathcal{A}$ has at most $2^{|U|}$ stable extensions. □

Theorem 17. *Given a negative AND-NOT BN f. Let U be a subset of vertices that intersects every positive cycle of $\text{ig}(f)$. Then f has at most $3^{|U|}$ complete trap spaces.*

Proof. For each assignement $m\colon U \mapsto \mathbb{B}_\star$, we build the transformed BN f^m of f as follows. For all $a \in U$, if $m(a) \neq \star$, then $f^m_a = m(a)$; otherwise, $f^m_a = \neg a$. For all $a \in \text{var}_f \setminus U$, $f^m_a = f_a$. Indeed, $\text{ig}(f^m)$ has no positive cycle, since U intersects all positive cycles of $\text{ig}(f)$, and the construction removes all predecessors of vertices in U on $\text{ig}(f)$ and only adds to $\text{ig}(f^m)$ a new negative cycle $a \xrightarrow{\ominus} a$ in the case $m(a) = \star$. Clearly, f^m is a negative AND-NOT BN. By Theorem 12, f^m has a unique complete trap space.

Note that the complete trap spaces of f agreeing with m are complete trap spaces of f^m. There are $3^{|U|}$ possible assignments w.r.t. U. Hence, f has at most $3^{|U|}$ complete trap spaces. □

Theorem 18. *Given an AF $\mathcal{A}$. Let U be a subset of vertices that intersects every positive cycle of $ag(\mathcal{A})$. Then $\mathcal{A}$ has at most $3^{|U|}$ complete extensions.*

Proof. It straightforwardly follows from Proposition 1, Theorem 1, and Theorem 17. □

Finally, we show a tighter upper bound for the number of stable extensions of an AF (Theorem 21) as if a subset of vertices intersects every positive cycle, it also intersects every positive cycle without a killing triple of the attack graph.

In an AND-NOT BN, every update function is only a conjunction of literals. Similar to negative AND-NOT BNs, an AND-NOT BN is uniquely determined by its influence graph.

Definition 7 ((Richard and Ruet, 2013))**.** *Given a signed directed graph G, a cycle C of G, and vertices u, v_1, v_2 of G. (u, v_1, v_2) is said to be a* delocalizing triple *of C when 1) v_1, v_2 are distinct vertices of C; 2) $(uv_1, \oplus)$ and $(uv_2, \ominus)$ are arcs of G that are not in C.*

Theorem 19 ((Veliz-Cuba et al., 2012))**.** *Let f be an AND-NOT BN. Assume that U_0 is a subset of vertices that intersects every positive cycle without a delocalizing triple. Then f has at most $2^{|U_0|}$ fixed points.*

Theorem 20. *Given a negative AND-NOT BN f. Let U_0 be a subset of vertices that intersects every positive cycle without a killing triple of ig(f). Then f has at most $2^{|U_0|}$ fixed points.*

Proof. We build from f a new BN f' as follows. For every positive cycle C in $\text{ig}(f)$ having a killing triple (u, v_1, v_2), let w be a subdivision of (u, v_1). Then we remove w from f and replace w by $\neg u$ in f_{v_1}. We always can do this because the in-degree and the out-degree of w both equal 1. Finally, we obtain f' that is an AND-NOT BN, but may not be a negative AND-NOT BN.

C can still be a cycle in $\text{ig}(f')$ or it becomes a new cycle with one vertex fewer in $\text{ig}(f')$. In any case, C still contains to a positive cycle C' in $\text{ig}(f')$. We have two cases for each C' as follows.

Case 1: $u \neq v_1$. We have that $(uv_2, \ominus)$ is an arc of $\text{ig}(f')$ but it does not belong to C'. $(uv_1, \oplus)$ is an arc of $\text{ig}(f')$. Since $(uv_1, \ominus)$ is not an arc of $\text{ig}(f)$, $(uv_1, \oplus)$ does not belong to C'. By definition, (u, v_1, v_2) is a delocalizing triple of C'.

Case 2: $u = v_1$. By a similar reasoning, we have that (u, v_1, v_2) is a delocalizing triple of C'. However, $\text{ig}(f')$ has a new positive cycle $v_1 \xrightarrow{\oplus} v_1$ that has no delocalizing triple in $\text{ig}(f')$. Of course, since $|w^-| = |w^+| = 1$, the positive cycle $w \xrightarrow{\ominus} v_1 \xrightarrow{\ominus} w$ has no killing triple in $\text{ig}(f)$. Hence, $U_0 \cap \{w, v_1\} \neq \emptyset$ in $\text{ig}(f)$. If $v_1 \in U_0$, then $v_1 \in U_0$ in $\text{ig}(f')$. If $v_1 \notin U_0$ and $w \in U_0$, then we replace w by v_1 in U_0. The size of U_0 does not change, but it does not miss any another positive cycle without a delocalizing triple because w cannot belong to any another positive cycle in $\text{ig}(f)$.

At the end, we always have a subset of vertices (say U_1) that intersects every positive cycle without a delocalizing triple in $\text{ig}(f')$ and $|U_1| = |U_0|$.

By the reduction results w.r.t. fixed points (Veliz-Cuba, 2011) and the fact that $(ww, \ominus)$ is not an arc of $\text{ig}(f)$, there is a bijection between the set of fixed points of f and that of f'. By Theorem 19, f' has at most $2^{|U_1|}$ fixed points. It follows that f has at most $2^{|U_0|}$ fixed points. □

Example 5. *Consider two signed directed graphs: G'_1 and G'_2 (see Figures 4(a) and 4(b), respectively). G'_1 (resp. G'_2) is the new graph transformed from G_1 (resp. G_2) of Example 4 by following the transformation shown in the proof of Theorem 20. In G_1, (a, d, c) is a killing triple of positive cycle $c \xrightarrow{\ominus} d \xrightarrow{\ominus} c$, and in G'_1, (a, d, c) becomes a delocalizing tripple of this positive cycle. In G_2, (c, c, a) is a killing triple of positive cycle $a \xrightarrow{\ominus} b \xrightarrow{\ominus} c \xrightarrow{\ominus} d \xrightarrow{\ominus} a$, and in G'_2, (c, c, a) becomes a delocalizing triple of this positive cycle. The new positive cycle $c \xrightarrow{\oplus} c$ in G'_2 has no delocalizing triple.*

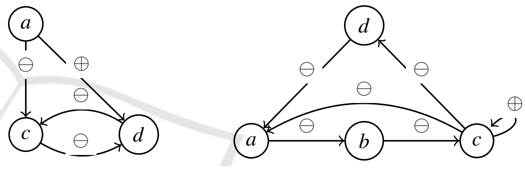


Figure 4: (a) G'_1 and (b) G'_2. These graphs are considered in Example 5.

Theorem 21. *Given an AF $\mathcal{A}$. Let U_0 be a subset of vertices that intersects every positive cycle without a killing triple of $ag(\mathcal{A})$. Then $\mathcal{A}$ has at most $2^{|U_0|}$ stable extensions.*

Proof. It straightforwardly follows from Theorem 20, Proposition 1, and Corollary 1. □

It is worth noting that Theorem 20 is also new in the BN theory.

6 CONCLUSION

In this work, we established the connection between AFs and BNs. More specifically, we showed that the attack graph of an AF coincides with the influence graph of its encoded BN, and preferred (resp. stable) extensions of the AF one-to-one correspond to minimal trap spaces (resp. fixed points) of the encoded BN. We defined a new concept of complete trap space and showed that complete extensions of the AF one-to-one correpond to complete trap spaces of the BN, and in particular the grounded extension of the AF one-to-one corresponds to the percolation of the whole-space trap space of the BN. We also showed a more general characterization of complete trap spaces in BNs. We then applied this connection to the graphical analysis of AFs: showing graphical conditions for properties on preferred, stable, and complete extensions. In particular, we showed new upper bounds based on positive feedback vertex sets for the numbers of stable, preferred, and complete extensions. An advantage of the above approach is that we only need to focus on dynamical properties of BNs when studying extensions in AFs, openning the great potential to obtain more improved theoretical results regarding the graphical analysis of AFs.

REFERENCES

Aracena, J. (2008). Maximum number of fixed points in regulatory Boolean networks. *Bull. Math. Biol.*, 70(5):1398–1409.

Azpeitia, E., Gutiérrez, S. M., Rosenblueth, D. A., and Zapata, O. (2024). Bridging abstract dialectical argumentation and Boolean gene regulation. *CoRR*, abs/2407.06106.

Baroni, P., Toni, F., and Verheij, B. (2020). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games: 25 years later. *Argument Comput.*, 11(1-2):1–14.

Baumann, R., Dvorák, W., Linsbichler, T., Strass, H., and Woltran, S. (2014). Compact argumentation frameworks. In *Proc. of ECAI*, pages 69–74. IOS Press.

Baumann, R. and Strass, H. (2013). On the maximal and average numbers of stable extensions. In *Proc. of TAFA*, pages 111–126. Springer.

Baumann, R. and Strass, H. (2015). Open problems in abstract argumentation. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, pages 325–339. Springer.

Baumann, R. and Ulbricht, M. (2021). On cycles, attackers and supporters - A contribution to the investigation of dynamics in abstract argumentation. In *Proc. of IJCAI*, pages 1780–1786. ijcai.org.

Caminada, M., Sá, S., Alcântara, J. F. L., and Dvorák, W. (2015). On the equivalence between logic programming semantics and argumentation semantics. *Int. J. Approx. Reason.*, 58:87–111.

Caminada, M. W. A. and Gabbay, D. M. (2009). A logical account of formal argumentation. *Stud Logica*, 93(2-3):109–145.

Charwat, G., Dvorák, W., Gaggl, S. A., Wallner, J. P., and Woltran, S. (2015). Methods for solving reasoning problems in abstract argumentation - A survey. *Artif. Intell.*, 220:28–63.

Coste-Marquis, S., Devred, C., and Marquis, P. (2005). Symmetric argumentation frameworks. In *Proc. of ECSQARU*, pages 317–328. Springer.

Dewoprabowo, R., Fichte, J. K., Gorczyca, P. J., and Hecher, M. (2022). A practical account into counting Dung's extensions by dynamic programming. In *Proc. of LPNMR*, pages 387–400. Springer.

Dimopoulos, Y., Dvorák, W., and König, M. (2024). Connecting abstract argumentation and Boolean networks. In *Proc. of COMMA*, pages 85–96. IOS Press.

Dimopoulos, Y. and Torres, A. (1996). Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2):209–244.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358.

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

Dunne, P. E., Dvorák, W., Linsbichler, T., and Woltran, S. (2015). Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228:153–178.

Elaroussi, M., Nourine, L., Radjef, M. S., and Vilmin, S. (2023). On the preferred extensions of argumentation frameworks: Bijections with naive sets. *Inf. Process. Lett.*, 181:106354.

Gaspers, S. and Li, R. (2019). Enumeration of preferred extensions in almost oriented digraphs. In *Proc. of MFCS*, pages 74:1–74:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Heyninck, J., Knorr, M., and Leite, J. (2024). Abstract dialectical frameworks are Boolean networks. In Dodaro, C., Gupta, G., and Martinez, M. V., editors, *Proc. of LPNMR*, pages 98–111. Springer.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22(3):437–467.

Klarner, H., Bockmayr, A., and Siebert, H. (2015). Computing maximal and minimal trap spaces of Boolean networks. *Nat. Comput.*, 14(4):535–544.

Kröll, M., Pichler, R., and Woltran, S. (2017). On the complexity of enumerating the extensions of abstract argumentation frameworks. In *Proc. of IJCAI*, pages 1145–1152. ijcai.org.

Nieves, J. C., Cortés, U., and Osorio, M. (2008). Preferred extensions as stable models. *Theory Pract. Log. Program.*, 8(4):527–543.

Nouioua, F. and Risch, V. (2012). A reconstruction of abstract argumentation admissible semantics into defaults and answer sets programming. In *Proc. of ICAART*, pages 237–242. SciTePress.

Obiedkov, S. and Sertkaya, B. (2023). Computing stable extensions of argumentation frameworks using formal concept analysis. In *Proc. of JELIA*, pages 176–191. Springer.

Paulevé, L. and Richard, A. (2012). Static analysis of Boolean networks based on interaction graphs: a survey. *Electron. Notes Theor. Comput. Sci.*, 284:93–104.

Pollock, J. L. (1987). Defeasible reasoning. *Cogn. Sci.*, 11(4):481–518.

Pollock, J. L. (1991a). Self-defeating arguments. *Minds Mach.*, 1(4):367–392.

Pollock, J. L. (1991b). A theory of defeasible reasoning. *Int. J. Intell. Syst.*, 6(1):33–54.

Richard, A. (2009). Positive circuits and maximal number of fixed points in discrete dynamical systems. *Discret. Appl. Math.*, 157(15):3281–3288.

Richard, A. (2010). Negative circuits and sustained oscillations in asynchronous automata networks. *Adv. Appl. Math.*, 44(4):378–392.

Richard, A. (2019). Positive and negative cycles in Boolean networks. *J. Theor. Biol.*, 463:67–76.

Richard, A. and Comet, J. (2007). Necessary conditions for multistationarity in discrete dynamical systems. *Discret. Appl. Math.*, 155(18):2403–2413.

Richard, A. and Ruet, P. (2013). From kernels in directed graphs to fixed points and negative cycles in Boolean networks. *Discret. Appl. Math.*, 161(7-8):1106–1117.

Rozum, J. C., Gómez Tejeda Zañudo, J., Gan, X., Deritei, D., and Albert, R. (2021). Parity and time reversal elucidate both decision-making in empirical models and attractor scaling in critical Boolean networks. *Sci. Adv.*, 7(29):eabf8124.

Schwab, J. D., Kühlwein, S. D., Ikonomi, N., Kühl, M., and Kestler, H. A. (2020). Concepts in Boolean network modeling: What do they all mean? *Comput. Struct. Biotechnol. J.*, 18:571–582.

Thimm, M., Cerutti, F., and Vallati, M. (2021). Skeptical reasoning with preferred semantics in abstract argumentation without computing preferred extensions. In *Proc. of IJCAI*, pages 2069–2075. ijcai.org.

Toulmin, S. (1958). *The Uses of Argument*. Cambridge University Press, Cambridge, England.

Trinh, G. V., Benhamou, B., Pastva, S., and Soliman, S. (2024a). Scalable enumeration of trap spaces in Boolean networks via answer set programming. In *Proc. of AAAI*, pages 10714–10722. AAAI Press.

Trinh, V. and Benhamou, B. (2024). Static analysis of logic programs via Boolean networks. *CoRR*, abs/2407.09015.

Trinh, V.-G., Benhamou, B., and Soliman, S. (2023). Trap spaces of Boolean networks are conflict-free siphons of their Petri net encoding. *Theor. Comput. Sci.*, 971:114073.

Trinh, V.-G., Benhamou, B., Soliman, S., and Fages, F. (2024b). Graphical conditions for the existence, unicity and number of regular models. In *Proc. of ICLP*, pages 175–186.

Trinh, V.-G., Park, K. H., Pastva, S., and Rozum, J. C. (2024c). Mapping the attractor landscape of Boolean networks. *bioRxiv*.

Ulbricht, M. (2021). On the maximal number of complete extensions in abstract argumentation frameworks. In *Proc. of KR*, pages 707–711.

Veliz-Cuba, A. (2011). Reduction of Boolean network models. *J. Theor. Biol.*, 289:167–172.

Veliz-Cuba, A., Buschur, K., Hamershock, R., Kniss, A., Wolff, E., and Laubenbacher, R. (2012). AND-NOT logic framework for steady state analysis of Boolean network models.

Yun, B., Croitoru, M., Vesic, S., and Bisquert, P. (2017). Graph theoretical properties of logic based argumentation frameworks: Proofs and general results. In *Proc. of GKR*, pages 118–138. Springer.