Independence and D-separation in Abstract Argumentation

Tjitze Rienstra[1], **Matthias Thimm**[1], **Kristian Kersting**[2], **Xiaoting Shao**[2]
[1]University of Koblenz-Landau, Germany
[2]TU Darmstadt, Germany
{rienstra, thimm}@uni-koblenz.de, {kersting,xiaoting.shao}@cs.tu-darmstadt.de

Abstract

We investigate the notion of *independence* in abstract argumentation, i. e., the question of whether the evaluation of one set of arguments is independent of the evaluation of another set of arguments, given that we already know the status of a third set of arguments. We provide a semantic definition of this notion and develop a method to discover independencies based on transforming an argumentation framework into a DAG on which we then apply the well-known *d-separation* criterion. We also introduce the *SCC Markov* property for argumentation semantics, which generalises the Markov property from the classical acyclic case and guarantees soundness of our approach.

1 Introduction

Efficient reasoning requires the ability to distinguish parts of a database that are relevant and irrelevant to a given query. Indeed, notions of (ir)relevance in various settings have been extensively studied in artificial intelligence (Koller and Halpern 1996; Kourousias and Makinson 2007; Borg and Straßer 2018). In probabilistic reasoning, for example, irrelevance is formalised as *conditional independence*. This is a fundamental notion in Bayesian networks, where conditional independencies are represented using directed acyclic graphs (DAGs) that encode (causal) relationships between variables. This idea is not limited to probabilistic reasoning, however. Darwiche (1997) showed that propositional knowledge can similarly be structured with DAGs, and that problems such as entailment, abduction, and diagnosis can benefit computationally from using such a representation.

In this paper, we study conditional independence in abstract argumentation. Our goal is to be able to answer the question: when are two sets of arguments A and B independent given a third set C? This notion of independence is defined relative to a given argumentation framework F and semantics σ, and is concerned with the *status* of the arguments involved. That is, A and B are independent given C if, once we know the status of the arguments in C, then knowing the status of the arguments in A does not tell us anything about the status of the arguments in B and vice versa. Our goal is to be able to answer questions about independence by examining the structure of the argumentation framework. This means that we can answer such questions without having to compute the labelings of the argumentation framework, which is often intractable.

While we study independence in abstract argumentation from a purely theoretical perspective, there are a number of practical applications. Generally speaking, knowledge about independence can be used to decompose queries into smaller sub-queries which may be executed separately. For example, if A and B are independent given C then, once the status of the elements of C is fixed, the status of the elements of A and of B have no bearing on each other and may therefore be computed separately. Our notion of independence is also useful in structured argumentation, where a claim is identified with the set of arguments whose conclusion equals the claim. Because we consider independence between sets of arguments, we can also reason about independence of between claims. Consider, for example, a disjunctive query $\phi \vee \psi$, which cannot in general be decomposed into a query for ϕ and for ψ, because it is possible that $\phi \vee \psi$ is skeptically accepted while neither ϕ nor ψ is skeptically accepted. However, if the sets of arguments with claim ϕ and ψ are independent, then decomposition is possible, because independence implies that $\phi \vee \psi$ is skeptically accepted if and only if either ϕ or ψ is skeptically accepted. Other argumentation-based applications where knowledge about independence may prove useful are persuasion, negotiation, decision making and explanation.

The method that we develop in this paper is based on the *d-separation* criterion. This is a well-known graph-theoretical criterion used to answer questions about independence on the basis of a causal graph or Bayesian network. Like the edges of a causal graph, attacks in an argumentation framework can be interpreted as a relationship of direct causal influence. An important difference, however, is that argumentation frameworks may contain cycles, while causal graphs are DAGs (acyclic). Because d-separation works only in combination with DAGs, we cannot apply it to an argumentation framework in a direct manner. Our solution is to first transform the argumentation framework to a DAG, called a *d-graph*, which captures the same independence information as the argumentation framework, but without using cycles. We can thus use d-separation in the d-graph of an argumentation framework to derive independence statements about its labelings, which requires only polynomial time. We prove that the independence statements derived like this are true for any semantics that is *SCC Markovian*, a new property we introduce for this very purpose. We show how this property is related to other known semantics properties, and prove that it is satisfied by (among others) the complete and preferred semantics but not by the stable and semi-stable semantics.

In summary, the contributions of this paper are as follows:

1. We introduce a semantical definition of (conditional) independence for abstract argumentation (Section 4).
2. We introduce the property *SCC Markovian* for abstract argumentation semantics and show that it is a sufficient condition for deriving independencies (Section 5).
3. We provide a syntactic criterion for deriving independencies using a transformation to an acyclic representation of the dependencies in an abstract argumentation framework and show its soundness (Section 6).

We provide necessary preliminaries about graphs and abstract argumentation in Section 2, present the necessary background on the concept of d-separation in Section 3, discuss related work an open issues in Section 7, and conclude in Section 8.

2 Preliminaries

A *directed graph* is a pair $G = (X, \to)$ where X is the set of *vertices* and $\to \subseteq X \times X$ the set of *edges*. We say that G is a *DAG* (directed acyclic graph) if G contains no directed cycles. In this paper we use directed graphs to represent sets of *variables* and relationships between variables. A variable is an object with an associated nonempty and finite set of possible values, called its *domain*. We denote variables using boldface letters $(\mathbf{x}, \mathbf{y}, \dots)$ and use $Dom(\mathbf{x})$ to denote the domain of $\mathbf{x}$. We use $\mathbf{X}$ to denote sets of variables.

A *valuation* of a set $\mathbf{X}$ of variables is a function V that maps every variable $\mathbf{x} \in \mathbf{X}$ to a value $V(\mathbf{x}) \in Dom(\mathbf{x})$. We denote by $\mathcal{V}(\mathbf{X})$ the set of all valuations of $\mathbf{X}$. A *belief state* over $\mathbf{X}$ is a subset of $\mathcal{V}(\mathbf{X})$ (we can think of a belief state as a proposition). Given a directed graph $G = (\mathbf{X}, \to)$, a valuation of (resp. belief state over) G is simply a valuation of (resp. belief state over) $\mathbf{X}$. Given a set $B \subseteq \mathbf{X}$ and a valuation $V \in \mathcal{V}(\mathbf{X})$ we denote by $V{\downarrow}B$ the restriction of V to B. Given a belief state $T \subseteq \mathcal{V}(\mathbf{X})$ we denote by $T{\downarrow}B$ the set $\{V{\downarrow}B \mid V \in T\}$. Given a directed graph $G = (\mathbf{X}, \to)$ we denote by $G{\downarrow}B$ the graph $(B, \to \cap B \times B)$.

An *argumentation framework* (abbreviated as *AF* in this paper) is usually defined as a directed graph whose vertices and edges represent arguments and attacks between arguments (Dung 1995). The three-valued labelling-based semantics for argumentation frameworks is based on *labelling* functions that map every argument to one of three labels: $\mathtt{I}$ for *in* (or *accepted*), $\mathtt{O}$ for *out* (or *rejected*) and $\mathtt{U}$ for *undecided* (Caminada and Gabbay 2009). We formalise this as follows. An *argument* is a variable $\mathbf{x}$ with a fixed three-valued domain $Dom(\mathbf{x}) = \{\mathtt{I}, \mathtt{O}, \mathtt{U}\}$. An AF is then simply a directed graph over a set of arguments, and a labelling of an AF is a valuation of the AF. To distinguish AFs from arbitrary directed graphs over variables we denote an AF as $F = (\mathbf{A}, \Rightarrow)$. To distinguish labelings from arbitrary valuations we use L to denote labelings. Given an AF $F = (\mathbf{A}, \Rightarrow)$ and arguments $\mathbf{x}, \mathbf{y} \in \mathbf{A}$ we say that $\mathbf{x}$ attacks $\mathbf{y}$ whenever $\mathbf{x} \Rightarrow \mathbf{y}$.

A *complete* labelling of an AF represents a position on argument acceptance where an argument is accepted whenever its attackers are rejected, and rejected whenever an attacker is accepted. More precisely, a labelling L of an AF $F = (\mathbf{A}, \Rightarrow)$ is complete if and only if for all $\mathbf{x} \in \mathbf{A}$, we have (1) $L(\mathbf{x}) = \mathtt{I}$ if and only if for all $\mathbf{y} \in \mathbf{A}$ such that $\mathbf{y} \Rightarrow \mathbf{x}$ we have $L(\mathbf{y}) = \mathtt{O}$; and (2) $L(\mathbf{x}) = \mathtt{O}$ if and only if there is a $\mathbf{y} \in \mathbf{A}$ such that $\mathbf{y} \Rightarrow \mathbf{x}$ and $L(\mathbf{y}) = \mathtt{I}$. Various additional criteria may be considered for a labelling to represent a reasonable position. A *semantics* σ maps every AF F to a set of labellings of F (i.e., a belief state over F) denoted $\mathcal{L}_\sigma(F)$, which consists of labellings that satisfy some set of criteria. The **co** (*complete*), **pr** (*preferred*), **gr** (*grounded*), **ss** (*semi-stable*) and **stb** (*stable*) semantics are defined by

Definition 1.

$$\mathcal{L}_{\textit{co}}(F) = \{L \in \mathcal{V}(F) \mid L \textit{ is a complete labelling of } F\},$$
$$\mathcal{L}_{\textit{pr}}(F) = \{L \in \mathcal{L}_{\textit{co}}(F) \mid \nexists L' \in \mathcal{L}_{\textit{co}}(F), L^{-1}(\mathtt{I}) \subset L'^{-1}(\mathtt{I})\},$$
$$\mathcal{L}_{\textit{gr}}(F) = \{L \in \mathcal{L}_{\textit{co}}(F) \mid \nexists L' \in \mathcal{L}_{\textit{co}}(F), L^{-1}(\mathtt{I}) \supset L'^{-1}(\mathtt{I})\},$$
$$\mathcal{L}_{\textit{ss}}(F) = \{L \in \mathcal{L}_{\textit{co}}(F) \mid \nexists L' \in \mathcal{L}_{\textit{co}}(F), L^{-1}(\mathtt{U}) \supset L'^{-1}(\mathtt{U})\}.$$
$$\mathcal{L}_{\textit{stb}}(F) = \{L \in \mathcal{L}_{\textit{co}}(F) \mid L^{-1}(\mathtt{U}) = \emptyset\}.$$

3 D-separation

A *causal graph* is a DAG $G = (\mathbf{X}, \to)$ with a specific interpretation: an edge from a variable $\mathbf{x}$ to a variable $\mathbf{y}$ means that $\mathbf{x}$ is a direct cause of $\mathbf{y}$. D-separation is a criterion used to identify *conditional independencies* implied by a causal graph. It allows us to determine whether two sets A, B of variables are independent given a third *observed* set C, by examining the structure of the causal graph (Pearl 2000).

Definition 2. *Let $G = (\mathbf{X}, \to)$ be a DAG. A* trail *in G is a loop-free, undirected (i.e., edge directions are ignored) path between two variables. If A, B, C are three disjoint sets of variables in G then A and B are said to be d-separated by C if every trail between every variable in A and in B is blocked by C. A trail is blocked by C if either:*

- *It contains a triple $\mathbf{x} \to \mathbf{z} \to \mathbf{y}$ or $\mathbf{x} \leftarrow \mathbf{z} \to \mathbf{y}$ such that $\mathbf{z} \in C$.*
- *It contains a triple $\mathbf{x} \to \mathbf{z} \leftarrow \mathbf{y}$ and neither $\mathbf{z}$ nor a descendant of $\mathbf{z}$ is in C.*

If A and B are d-separated by C we also write $D_G(A, B \mid C)$, and if not we write $\neg D_G(A, B \mid C)$. If we speak of d-separation of individual variables we mean d-separation of the singleton sets containing these variables.

Example 1. *Consider the causal graph shown in figure 1. The "chain" structure $\boldsymbol{a} \to \boldsymbol{c} \to \boldsymbol{d}$ indicates that $\boldsymbol{a}$ is indirectly a cause of $\boldsymbol{d}$. While $\boldsymbol{a}$ and $\boldsymbol{d}$ are not d-separated by the empty set, they are d-separated by $\boldsymbol{c}$.*

The "fork" structure $\boldsymbol{c} \leftarrow \boldsymbol{b} \to \boldsymbol{e}$ indicates that $\boldsymbol{b}$ is a common cause of $\boldsymbol{c}$ and $\boldsymbol{e}$. Here, $\boldsymbol{c}$ and $\boldsymbol{e}$ are not d-separated by the empty set, but they are d-separated by $\boldsymbol{b}$. Intuitively, a dependency between $\boldsymbol{c}$ and $\boldsymbol{e}$ may arise due to the common cause $\boldsymbol{b}$, but observing $\boldsymbol{b}$ renders $\boldsymbol{c}$ and $\boldsymbol{e}$ independent.

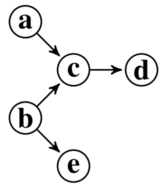


Figure 1: A Causal Graph

The "collider" structure $\boldsymbol{a} \to \boldsymbol{c} \leftarrow \boldsymbol{b}$ *indicates that* $\boldsymbol{c}$ *is a common effect of* $\boldsymbol{a}$ *and* $\boldsymbol{b}$*. Here,* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *are d-separated by the empty set but not by* $\boldsymbol{c}$*. Intuitively, while* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *are independent, observing* $\boldsymbol{c}$ *may make them dependent, as refuting one cause may make the other cause more probable. Since* $\boldsymbol{d}$ *is also a common cause of* $\boldsymbol{a}$ *and* $\boldsymbol{b}$*, it similarly holds that* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *are not d-separated by* $\boldsymbol{d}$*.*

Now consider $\boldsymbol{a}$ *and* $\boldsymbol{e}$*. They are d-separated by the empty set, as the trail between* $\boldsymbol{a}$ *and* $\boldsymbol{e}$ *contains the triple* $\boldsymbol{a} \to \boldsymbol{c} \leftarrow \boldsymbol{b}$ *and neither* $\boldsymbol{c}$ *nor a descendant of* $\boldsymbol{c}$ *is in* $\emptyset$*. They are not d-separated by* $\boldsymbol{c}$ *however. They are d-separated again given* $\{\boldsymbol{c}, \boldsymbol{b}\}$*, as the trail between* $\boldsymbol{a}$ *and* $\boldsymbol{e}$ *contains the triple* $\boldsymbol{c} \leftarrow \boldsymbol{b} \to \boldsymbol{e}$ *and* $\boldsymbol{b}$ *is in* $\{\boldsymbol{c}, \boldsymbol{b}\}$*.*

Bayesian networks are causal graphs used as compact representations of probability distributions. Given a Bayesian network $G = (\mathbf{X}, \to)$, a probability distribution P over $\mathbf{X}$ is said to satisfy the *local Markov condition* with respect to G if, according to P, each variable in G is probabilistically independent of its nondescendants given its parents (see Definition 3 below). For Bayesian networks, d-separation is sound in the sense that, if P satisfies the local Markov condition with respect to G, then then if A and B are d-separated by C in G, then A and B are probabilistically independent given C in P (Pearl 2000). More generally, d-separation is sound with respect to any *dependency model* satisfying certain properties. A dependency model over a set $\mathbf{X}$ of variables is a three-place relation I over disjoint subsets of $\mathbf{X}$. The elements of I are understood as statements about conditional independence. That is, $(A, B, C) \in I$ holds when A and B are independent given C. For convenience we write $I(A, B \mid C)$ instead of $(A, B, C) \in I$ and $\neg I(A, B \mid C)$ instead of $(A, B, C) \notin I$. A dependency model I over $\mathbf{X}$ that satisfies the following set of axioms (for all disjoint subsets A, B, C, D of $\mathbf{X}$) is called a *semi-graphoid*:

- Symmetry: If $I(A, B \mid C)$ then $I(B, A \mid C)$.
- Decomposition: If $I(A, B \cup D \mid C)$ then $I(A, B \mid C)$.
- Weak Union: If $I(A, B \cup D \mid C)$ then $I(A, D \mid C \cup B)$.
- Contraction: If $I(A, D \mid C \cup B)$ and $I(A, B \mid C)$ then $I(A, B \cup D \mid C)$.

These axioms represent properties that any dependency model can reasonably be expected to satisfy (see, e.g., (Pearl 1989) for a justification). Indeed, dependency models defined by probabilistic independence relationships are semi-graphoids, but various other types of independence that have been considered are also semi-graphoids. The local Markov condition for dependency models is defined as follows.

Definition 3. *Given a DAG* $G = (\mathbf{X}, \to)$ *and two variables* $\mathbf{x}, \mathbf{y} \in \mathbf{X}$ *we say that* $\mathbf{x}$ *is a* parent *of* $\mathbf{y}$ *if* $\mathbf{x} \to \mathbf{y}$*; that* $\mathbf{x}$ *is a* descendant *of* $\mathbf{y}$ *if* $\mathbf{x} = \mathbf{y}$ *or a directed path from* $\mathbf{y}$ *to* $\mathbf{x}$ *exists; and that* $\mathbf{x}$ *is a* nondescendant *of* $\mathbf{y}$ *if* $\mathbf{x}$ *is not a descendant or parent of* $\mathbf{y}$*. We denote the parents and nondescendants of* $\mathbf{x}$ *by* $Pa_G(\mathbf{x})$ *and* $ND_G(\mathbf{x})$*, respectively.*

Definition 4. *A dependency model* I *over* $\mathbf{X}$ *satisfies the* local Markov condition *with respect to a DAG* $G = (\mathbf{X}, \to)$ *if and only if*

$$\forall \mathbf{x} \in \mathbf{X} : I(\{\mathbf{x}\}, ND_G(\{\mathbf{x}\}) \mid Pa_G(\{\mathbf{x}\})). \tag{1}$$

The d-separation criterion is sound with respect to any DAG G and semi-graphoid I over the variables of G, provided that I satisfies the local Markov condition with respect to G. We make use of this fact, which was proven in (Verma and Pearl 1990, Theorem 19), later on in this paper.

Theorem 1. *Let* $G = (\mathbf{X}, \to)$ *be a DAG. If* I *is a semi-graphoid over* $\mathbf{X}$ *that satisfies the local Markov condition with respect to* G *then for every disjoint sets* $A, B, C \subseteq \mathbf{X}$ *we have that* $D_G(A, B \mid C)$ *implies* $I(A, B \mid C)$*.*

4 Conditional Independence in Abstract Argumentation

We now define a notion of conditional independence for use in abstract argumentation. The definition we use is based on the definition of conditional independence in propositional logic due to (Darwiche and Pearl 1994; Darwiche 1997). Intuitively, given a belief state T over a set $\mathbf{X}$ of variables, and disjoint subsets A, B, C of $\mathbf{X}$, A and B are independent given C if, once we know the values of C then knowing the values of A provides no information about B and vice versa.

First some auxiliary definitions. Let $\mathbf{X}$ be a set of variables. We say that a valuation V_A of some subset A of $\mathbf{X}$ is *consistent* with a belief state T over $\mathbf{X}$ if for some $V \in T$ we have $V{\downarrow}A = V_A$. Given two valuations V_A and V_B of disjoint subsets A and B of $\mathbf{X}$ we denote by $V_A \cup V_B$ the union of the two valuations. Conditional independence with respect to a belief state is defined as follows.

Definition 5. *Let* T *be a belief state over* $\mathbf{X}$*. Given disjoint subsets* A, B, C *of* $\mathbf{X}$*, we say that* A and B are independent given C in T *if, for all* $V_A \in \mathcal{V}(A), V_B \in \mathcal{V}(B), V_C \in \mathcal{V}(C)$*, consistency of* $V_C \cup V_A$ *and* $V_C \cup V_B$ *in* T *implies consistency of* $V_C \cup V_A \cup V_B$ *in* T*. We denote by* I_T *the dependency model over* $\mathbf{X}$ *defined by*

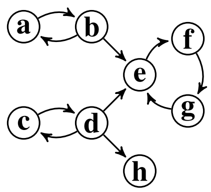
$$I_T(A, B \mid C) \leftrightarrow A \textit{ and } B \textit{ are independent given } C \textit{ in } T.$$

We first note that a dependency model defined by a belief state is a semi-graphoid.

Proposition 1. *Let* $\mathbf{X}$ *be a set of variables. For every belief state* T *over* $\mathbf{X}$ *it holds that* I_T *is a semi-graphoid.*

Given an AF F and semantics σ, we will abbreviate the dependency model $I_{\mathcal{L}_\sigma(F)}$ as I_F^σ. This dependency model captures the independencies that hold in the evaluation of F under the σ semantics. As can be seen by inspecting Definition 5, if the number of σ labellings of F is zero or one, then every independence statement holds trivially in I_F^σ. Thus,

F:



$\mathcal{L}_{pr}(F)$

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**
I	O	I	O	U	U	U	I
I	O	O	I	O	I	O	O
O	I	I	O	O	I	O	I
O	I	O	I	O	I	O	O

$\mathcal{L}_{st}(F)$

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**
I	O	O	I	O	I	O	O
O	I	I	O	O	I	O	I
O	I	O	I	O	I	O	O

Figure 2: An AF and its preferred and stable labellings

a dependency model I_F^σ is informative only if F possesses more than one σ labelling. We therefore only consider combinations of AFs and semantics where there is more than one labelling. In particular, we do not consider the grounded semantics.

Our goal is to define a method to infer, given an AF F and semantics σ, the independencies that hold in I_F^σ. We want to infer these independencies without having to compute the σ labellings of F, which is intractable under all semantics considered in this paper (Kröll, Pichler, and Woltran 2017). First we look at an example where we determine independencies in a direct manner, by examining the labellings of an AF and checking which independence statements are true.

Example 2. *Consider the AF F and the set $\mathcal{L}_{pr}(F)$ of preferred labellings shown in Figure 2. We have that:*

1. ***a** and **e** are not independent, since $(\mathbf{a}\colon \mathtt{O})$ and $(\mathbf{e}\colon \mathtt{U})$ are consistent but $(\mathbf{a}\colon \mathtt{O}, \mathbf{e}\colon \mathtt{U})$ is not. They do become independent after observing **b**. Thus, we have:*

$$\neg I_F^{pr}(\mathbf{a}, \mathbf{e} \mid \emptyset) \qquad I_F^{pr}(\mathbf{a}, \mathbf{e} \mid \mathbf{b})$$

2. *$\mathbf{e}$ and $\mathbf{h}$ are not independent. They do become independent after observing $\mathbf{d}$:*

$$\neg I_F^{pr}(\mathbf{e}, \mathbf{h} \mid \emptyset) \qquad I_F^{pr}(\mathbf{e}, \mathbf{h} \mid \mathbf{d})$$

3. *the sets $\{\mathbf{a}, \mathbf{b}\}$ and $\{\mathbf{c}, \mathbf{d}\}$ are independent, but they are not independent if we observe $\mathbf{e}$, since $(\mathbf{e}\colon \mathtt{O}, \mathbf{a}\colon \mathtt{I}, \mathbf{b}\colon \mathtt{O})$ and $(\mathbf{e}\colon \mathtt{O}, \mathbf{c}\colon \mathtt{I}, \mathbf{d}\colon \mathtt{O})$ are consistent but $(\mathbf{e}\colon \mathtt{O}, \mathbf{a}\colon \mathtt{I}, \mathbf{b}\colon \mathtt{O}, \mathbf{c}\colon \mathtt{I}, \mathbf{d}\colon \mathtt{O})$ is not. They are similarly not independent if we observe $\mathbf{f}$ or $\mathbf{g}$. Thus, we have:*

$$I_F^{pr}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \emptyset) \qquad \neg I_F^{pr}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \mathbf{f})$$
$$\neg I_F^{pr}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \mathbf{e}) \qquad \neg I_F^{pr}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \mathbf{g})$$

4. *$\mathbf{a}$ and $\mathbf{h}$ are independent, but they are not independent if we observe $\mathbf{e}$. This is because $(\mathbf{e}\colon \mathtt{O}, \mathbf{a}\colon \mathtt{I})$ and $(\mathbf{e}\colon \mathtt{O}, \mathbf{h}\colon \mathtt{I})$ are consistent but $(\mathbf{e}\colon \mathtt{O}, \mathbf{a}\colon \mathtt{I}, \mathbf{h}\colon \mathtt{I})$ is not. If we additionally observe $\mathbf{d}$ then they are independent again:*

$$I_F^{pr}(\mathbf{a}, \mathbf{h} \mid \emptyset) \quad \neg I_F^{pr}(\mathbf{a}, \mathbf{h} \mid \mathbf{e}) \quad I_F^{pr}(\mathbf{a}, \mathbf{h} \mid \{\mathbf{d}, \mathbf{e}\})$$

We can already see that the independencies in this example can be inferred from the structure of the AF similarly to how we inferred independencies using d-separation in Example 1. We have, for instance, a chain structure $\mathbf{a} \rightarrow \mathbf{b} \rightarrow \mathbf{e}$, where **a** and **e** become independent after observing **b**. There is also a fork structure $\mathbf{e} \leftarrow \mathbf{d} \rightarrow \mathbf{h}$, where **e** and **h** become independent after observing **d** (a common cause of **e** and **h**). We also have a collider structure $\mathbf{b} \rightarrow \mathbf{e} \leftarrow \mathbf{d}$, where $\{\mathbf{a}, \mathbf{b}\}$ and $\{\mathbf{c}, \mathbf{b}\}$ are unconditionally independent but not independent after observing **e** (a common effect of **b** and **d**).

Unfortunately, however, we cannot apply d-separation to the structure of an AF directly. The reason is that the soundness of d-separation as stated by Theorem 1 holds only DAGs, and AFs are not in general DAGs. Our approach will therefore be as follows. Given an AF F and semantics σ, we first transform F into a DAG which we call the *d-graph* of F. We then apply d-separation to the d-graph of F to infer independencies that hold in I_F^σ. Before presenting this transformation we consider a property that a semantics must satisfy in order for this approach to be sound.

5 The SCC Markov Principle

The independencies that hold in the evaluation of an AF depend on the semantics that we use. While Example 2 is based on the preferred semantics, it can be checked that the same independencies hold under the complete semantics. Under the stable semantics things are different, however. Consider again the AF F shown in Figure 2 and consider the set $\mathcal{L}_{st}(F)$ of stable labellings. While the sets $\{\mathbf{a}, \mathbf{b}\}$ and $\{\mathbf{c}, \mathbf{d}\}$ are unconditionally independent under the preferred semantics, they are not unconditionally independent under the stable semantics, nor under the semi-stable semantics, which coincides with the stable semantics in this case.

This behaviour is due to the resistance of these semantics to have arguments undecided. Because of this, either **b** or **d** must be accepted, and this leads to a dependency between the two sets, as observing that **b** is rejected implies that **d** is accepted and vice versa. If we interpret attacks, like edges in a causal graph, as a relationship of direct causal influence, then this behaviour is strange. Under this interpretation, $\{\mathbf{a}, \mathbf{b}\}$ and $\{\mathbf{c}, \mathbf{d}\}$ should indeed be unconditionally independent, as the two sets have no common cause and none of the common effects **e**, **f** or **g** are observed.

We now formally characterise the class of semantics that are consistent with the interpretation of attacks as a relationship of direct causal influence. We then show that the preferred and complete semantics belong to this class, but the stable and semi-stable semantics do not. First we consider a generalisation of the local Markov condition that is well-defined for graphs with cycles. The idea is based on the observation that every graph with cycles possess an acyclic structure, which is formed by the graph's *strongly connected components* (SCCs) and their interactions.

Definition 6. *The set of* SCCs *(*strongly connected components*) of a graph $G = (\mathbf{X}, \rightarrow)$, denoted $\mathcal{S}(G)$, contains the equivalence classes induced by the* path equivalence relation *$\sim_G$ over $\mathbf{X}$ defined by $\mathbf{x} \sim_G \mathbf{y}$ iff $\mathbf{x} = \mathbf{y}$ or there is a directed path from $\mathbf{x}$ to $\mathbf{y}$ and $\mathbf{y}$ to $\mathbf{x}$.*

Like parents and nondescendants of individual variables we also define parents and nondescendants of SCCs. Note

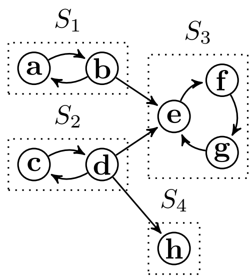


Figure 3: An AF with SCCs highlighted

that we define a parent of an SCC S to be a parent of an element of S that is itself not a member of S. This is also sometimes referred to as an *outparent*.

Definition 7. *Let G be a graph and let S be an SCC of G. A* parent *of S is a parent of an element of S that is itself not a member of S. A* descendant *of S is an element of S or a variable $\mathbf{x}$ such that directed path from an element of S to $\mathbf{x}$ exists. A* nondescendant *of S is any variable that is not a descendant or parent of S. We denote the parents and nondescendants of S by $Pa_G(S)$ and $ND_G(S)$, respectively.*

Example 3. *Let F be the AF shown in Figure 3, which is the same AF as shown in Figure 2 with SCCs enclosed in dotted rectangles. We have:*

$$\begin{array}{ll} Pa_F(S_1) = \emptyset & ND_F(S_1) = \{\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{h}\} \\ Pa_F(S_2) = \emptyset & ND_F(S_2) = \{\boldsymbol{a}, \boldsymbol{b}\} \\ Pa_F(S_3) = \{\boldsymbol{b}, \boldsymbol{d}\} & ND_F(S_3) = \{\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{h}\} \\ Pa_F(S_4) = \{\boldsymbol{d}\} & ND_F(S_4) = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}\} \end{array}$$

The *SCC Markov condition* states that every SCC is independent of its nondescendants given its outparents.

Definition 8. *A dependency model I over $\mathbf{X}$ satisfies the* SCC-Markov condition *with respect to a (possibly cyclic) graph $G = (\mathbf{X}, \rightarrow)$ if and only if*

$$\forall \mathbf{S} \in \mathcal{S}(G) : I(S, ND_G(S) \mid Pa_G(S)). \tag{2}$$

Note that the SCC-Markov condition implies nothing about independencies among elements of the same SCC. Hence, nothing is implied about independencies among the elements of a cycle, as such elements belong to the same SCC. Furthermore, if G is a DAG then every SCC of G is singleton set $\{\mathbf{x}\}$ whose parents coincide with the parents of $\mathbf{x}$. This implies:

Proposition 2. *If G is a DAG then a dependency model I over G satisfies the local Markov condition with respect to G if and only if it satisfies the SCC Markov condition with respect to G.*

We define an *SCC Markovian* semantics as follows.

Definition 9. *A semantics σ is* SCC Markovian *if for every AF F, I_F^σ satisfies the SCC Markov condition with respect to F.*

Which semantics are SCC Markovian? We first observe that the stable and semi-stable semantics are not.

Example 4. *Figure 3 depicts the same AF as Figure 2 but with SCCs enclosed in dotted rectangles. The set $\mathcal{L}_{st}(F)$ of stable labellings of F are shown on the right in Figure 2. For the stable semantics to be SCC Markovian, I_{st}^F must satisfy the SCC Markov condition with respect to F. For example, the SCC $\{\mathbf{c}, \mathbf{d}\}$ must be independent of its parents given its nondescendants, i.e., $I_{st}^F(\{\mathbf{c}, \mathbf{d}\}, \{\mathbf{a}, \mathbf{b}\} \mid \emptyset)$. However, $\{\mathbf{c}\colon \mathtt{I}, \mathbf{d}\colon \mathtt{O}\}$ and $\{\mathbf{a}\colon \mathtt{I}, \mathbf{b}\colon \mathtt{O}\}$ are consistent in $\mathcal{L}_{st}(F)$ but $\{\mathbf{c}\colon \mathtt{I}, \mathbf{d}\colon \mathtt{O}, \mathbf{a}\colon \mathtt{I}, \mathbf{b}\colon \mathtt{O}\}$ is not. We thus have $\neg I_{st}^F(\{\mathbf{c}, \mathbf{d}\}, \{\mathbf{a}, \mathbf{b}\} \mid \emptyset)$ Therefore, the stable semantics is not SCC Markovian. Because the stable and semi-stable labellings of F coincide, this example also demonstrates that the semi-stable semantics is not SCC Markovian.*

The complete and preferred semantics, however, are SCC Markovian. To prove this we consider two general semantic principles and show that their combination is a sufficient condition for SCC Markovianness. These are *SCC decomposability* and *Universality*. These principles are satisfied by the complete and preferred semantics as well as a number of other semantics.

SCC decomposability was introduced by (Baroni et al. 2014), where it was called *full decomposability with respect to SCC partitioning*. To define it we need to define the notion of *AF with input*, which is an AF F together with a set of *input arguments* which may attack F via a provided *input attack relation*, as well as a labelling of the input arguments.

Definition 10. *An* AF with input *is a tuple $(F, B_{in}, L_{in}, \Rightarrow_{in})$ where $F = (\mathbf{A}, \Rightarrow)$ is an AF, B_{in} a set of* input arguments *such that $\mathbf{A} \cap B_{in} = \emptyset$, $L_{in} \in \mathcal{L}(B_{in})$ an* input labelling*, and $\Rightarrow_{in} \subseteq B_{in} \times \mathbf{A}$ an* input attack relation*.*

A *local function* assigns to each AF with input a set of labellings of the AF.

Definition 11. *A* local function *Z assigns to every AF with input $(F, B_{in}, L_{in}, \Rightarrow_{in})$ a set $Z(F, B_{in}, L_{in}, \Rightarrow_{in}) \subseteq \mathcal{L}(F)$.*

A semantics σ is SCC decomposable if it is *represented* by some local function. A local function Z represents a semantics σ if, for every AF F, the global computation of the σ labellings of F coincides with the per-SCC local computation according to Z.

Definition 12. *A semantics σ is* represented *by a local function Z if and only if for every AF $F = (\mathbf{A}, \Rightarrow)$ we have that $L \in \mathcal{L}_\sigma(F)$ if and only if $\forall S \in \mathcal{S}(F)$,*

$$L{\downarrow}S \in Z(F{\downarrow}S, Pa_F(S), L{\downarrow}Pa_F(S), \Rightarrow \cap Pa_F(S) \times S).$$

We say that σ is SCC decomposable *if and only if σ is represented by some local function.*

Universality simply states that every AF has at least one labelling.

Definition 13. *A semantics σ is* universal *if and only if for each AF F, $\mathcal{L}_\sigma(F) \neq \emptyset$.*

Any semantics that is both SCC decomposable and Universal is SCC Markovian.

Theorem 2. *If σ is SCC decomposable and Universal then σ is SCC Markovian.*

	SCC Decomposable	Universal	SCC Markovian
Complete	Yes	Yes	Yes
Preferred	Yes	Yes	Yes
Stable	Yes	No	No
Semi-stable	No	Yes	No

Table 1: Principles under different semantics

Proof. Follows from lemma 1 and 2 below. □

Note that the other direction of this implication does not hold. To see why, consider the semantics that evaluates single-SCC AFs using preferred semantics and all other AFs using complete semantics. This semantics is SCC Markovian and universal but not SCC decomposable. An example of a semantics that is SCC Markovian and SCC decomposable but not universal is the semantics that maps every AF to the empty set of labellings.

The complete and preferred semantics are both universal and SCC decomposable (Baroni et al. 2014) and hence SCC Markovian. We already saw that the stable and semi-stable are not SCC Markovian, which means that they must violate either universality or SCC decomposability. Indeed, the stable semantics is not universal, while the semi-stable semantics is not SCC decomposable (Baroni et al. 2014). Table 1 provides an overview of which of the semantics considered in this paper satisfy the principles discussed here.

In the remainder of this section, which may be skipped upon first reading, we present the two lemmas referred to in the proof of Theorem 2. In these lemmas we make use of what we call a *causal theory*, which is a graph G together with an *influence function* for each SCC of G. An influence function for an SCC S assigns to every valuation of the parents of S a *non-empty* set of valuations of S. The belief state represented by a causal theory is the belief state that is consistent with these assignments.[1]

Definition 14. *A* causal theory *is a pair (G, Φ) where G is a graph with SCCs $\{S_1, \ldots, S_n\}$ and $\Phi = \{\Phi_1, \ldots, \Phi_n\}$ is a set of* influence functions *where, for every $i \in 1 \ldots n$, $\Phi_i : \mathcal{V}(Pa_G(S_i)) \to (2^{\mathcal{V}(S_i)} \setminus \emptyset)$. A causal theory (G, Φ)* represents *the belief state T defined by*

$$V \in T \leftrightarrow \textit{for } i = 1 \ldots n, V{\downarrow}S_i \in \Phi_i(V{\downarrow}Pa_G(S_i)). \quad (3)$$

The non-emptiness condition for influence functions prevents the ability to express dependencies among parents of an SCC. To see why, note that $\Phi_i(V) = \emptyset$ would imply inconsistency of V, which may result in dependencies among the parents of S_i even if there are no edges between these parents.[2] The first lemma provides a link between causal theories and the SCC Markov condition.

[1] In the acyclic case, causal theories are similar to *symbolic causal networks* due to (Darwiche and Pearl 1994). Symbolic causal networks use sets of formulas, called *micro theories*, instead of influence functions. They furthermore support *exogenous variables*, which we do not need and thus omit.

[2] A condition similar to non-emptiness applies to the micro theories of a symbolic causal network (Darwiche and Pearl 1994).

Lemma 1. *For every belief state T we have that I_T satisfies the SCC Markov condition with respect to G if and only if some causal theory (G, Φ) represents T.*

Proof. (If) Suppose (G, Φ) represents T. Let $S \in \mathcal{S}(G)$, $ND = ND_F(S)$, $Pa = Pa_F(S)$. We need to prove that we have $I_T(S, ND \mid Pa)$. Let $V_{ND} \in \mathcal{V}(ND)$, $V_{Pa} \in \mathcal{V}(Pa)$, and $V_S \in \mathcal{V}(S)$. Suppose $V_{Pa} \cup V_S$ and $V_{Pa} \cup V_{ND}$ are consistent in T. Then there are valuations $V^1, V^2 \in T$ such that $V^1{\downarrow}Pa \cup S = V_{Pa} \cup V_S$ and $V^2{\downarrow}Pa \cup ND = V_{Pa} \cup V_{ND}$. Since (G, Φ) represents T we also have, for all $S' \in \mathcal{S}(F)$, $V^1{\downarrow}S' \in \Phi_{S'}(V^1{\downarrow}Pa_G(S'))$ and $V^2{\downarrow}S' \in \Phi_{S'}(V^2{\downarrow}Pa_G(S'))$. Now define V^3 by $V^3{\downarrow}S \cup Dsc_F(S) = V^1{\downarrow}S \cup Dsc_F(S)$ and $V^3{\downarrow}ND_F(S) \cup Pa_F(S) = V^2{\downarrow}ND_F(S) \cup Pa_F(S)$. Then for all $S \in \mathcal{S}(G)$ we have $V^3{\downarrow}S \in \Phi_S(V^3{\downarrow}Pa_F(S))$. Because (G, Φ) represents T it now follows that $V^3 \in \mathcal{L}_\sigma(F)$ and hence that $V_{Pa} \cup V_S \cup V_{ND}$ is consistent. We thus have $I_T(S, ND \mid Pa)$.

(Only If) Let T be a belief state such that I_T satisfies the SCC Markov condition with respect to G. Let $\{S_1, \ldots, S_n\} = \mathcal{S}(G)$ and define Φ_{S_i} by

$$\Phi_{S_i}(V_{Pa}) = \{V{\downarrow}S_i \mid V \in T, V{\downarrow}Pa_F(S_i) = V_{Pa}\}, \quad (4)$$

if V_{Pa} is consistent in T, and $\Phi_{S_i}(V_{Pa}) = \mathcal{L}(S_i)$, otherwise. We now prove that (3) holds. The $\to$ direction is easy. For the $\leftarrow$ direction, let V be a valuation s.t., for $i = 1 \ldots n$, $V{\downarrow}S_i \in \Phi_{S_i}(V{\downarrow}Pa_F(S_i))$. Assume w.l.o.g. that $\{S_1, \ldots, S_n\}$ is ordered such that if $i < j$ then $S_i \subseteq ND_G(S_j) \cup Pa_G(S_j)$. We prove by induction on i that $V{\downarrow}S_1 \cup \ldots \cup S_i$ is consistent in T for all $i \leq n$. For the base case we have $V{\downarrow}S_1 \in \Phi_{S_1}(\emptyset)$. Using (4) it then follows that $V{\downarrow}S_1$ is consistent in T. For the inductive case, assume that $V{\downarrow}\{S_1, \ldots, S_{i-1}\}$ is consistent in T. Since $S_1 \cup \ldots \cup S_{i-1} \subseteq ND_G(S_i) \cup Pa_G(S_i)$, the fact that I_T satisfies the SCC Markov condition with respect to G implies that S_i is independent of $S_1 \cup \ldots \cup S_{i-1} \setminus Pa_G(S_i)$ given $Pa_G(S_i)$. This implies that $V{\downarrow}\{S_1, \ldots, S_i\}$ is consistent in T. By induction it follows that $V{\downarrow}\{S_1, \ldots, S_n\}$ is consistent in T and hence that $V \in T$. Hence condition (3) is satisfied which means that (G, Φ) represents T. □

The second lemma states that, if σ is an SCC decomposable and universal semantics, then for every AF F we can construct a causal theory (F, Φ) that represents $\mathcal{L}_\sigma(F)$.

Lemma 2. *Let σ be an SCC decomposable and universal semantics. Let Z_σ be the local function that represents σ. Let $F = (\mathbf{A}, \Rightarrow)$ be an AF with SCCs $\{S_1, \ldots, S_n\}$. Let $(F, \{\Phi_1, \ldots, \Phi_n\})$ be the causal theory where*

$$\Phi_i(L) = Z_\sigma(F{\downarrow}S_i, Pa_F(S_i), L, \Rightarrow \cap Pa_F(S_i) \times S_i).$$

Then (F, Φ) represents $\mathcal{L}_\sigma(F)$.

Proof. Follows directly from Definitions 12 and 14. □

6 The D-graph Approach

We now show how to derive independencies from the structure of an AF by transforming the AF into what we call a *d-graph*. After presenting the transformation itself we discuss the soundness, completeness and complexity of the approach. Intuitively, the d-graph transformation is based on the principle of replacing cyclical dependencies with dependencies represented using extra *latent common cause* variables. If a graph contains, for example, a cycle $\mathbf{a} \leftrightarrow \mathbf{b}$, then the resulting d-graph contains a fork structure $\mathbf{a} \leftarrow \mathbf{s} \rightarrow \mathbf{b}$, where $\mathbf{s}$ is an extra variable representing a common cause for $\mathbf{a}$ and $\mathbf{b}$. This common cause $\mathbf{s}$ is hypothetical, used purely to account for the dependency between $\mathbf{a}$ and $\mathbf{b}$, and therefore treated as unobservable. As long as $\mathbf{s}$ is not observed, the two structures (cycle and fork) represent the same independence information, because the fork structure ensures that $\mathbf{a}$ and $\mathbf{b}$ are d-separated only if $\mathbf{s}$ is observed.

The transformation of a graph G to the d-graph G^* consists of three steps, applied separately to each SCC S_i:

1. Remove all edges between elements of S_i.
2. Add an extra latent common cause variable $\mathbf{s}_i$ and an edge from $\mathbf{s}_i$ to every element of S_i.
3. For every parent $\mathbf{x}$ of S_i, replace the edge from $\mathbf{x}$ to S_i with an edge from $\mathbf{x}$ to $\mathbf{s}_i$.

The result is a DAG since step 1 removes all cycles and steps 2 and 3 do not introduce new cycles. The following definition describes the d-graph transformation more consisely.

Definition 15. *Let $G = (\mathbf{X}, \rightarrow)$ be a graph with SCCs $\mathcal{S}(G) = \{S_1, \ldots, S_n\}$. The* d-graph *of G is a DAG $G^* = (\mathbf{X} \cup \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}, \rightarrow')$ where $\mathbf{x} \rightarrow' \mathbf{y}$ iff for some $i \in 1, \ldots, n$ either:*

- *$\mathbf{x} \in Pa_G(S_i)$ and $\mathbf{y} = \mathbf{s}_i$, or*
- *$\mathbf{x} = \mathbf{s}_i$ and $\mathbf{y} \in S_i$.*

Let us look at an example. Let F be the AF shown in Figure 3. The d-graph F^* is shown in Figure 4. In this figure we have highlighted the original SCCs with dotted rectangles. Note that, while the edges in F represent attacks, the edges in F^* represent arbitrary relations of direct influence. Consider the cycle in F contained in the SCC $S_1 = \{\mathbf{a}, \mathbf{b}\}$. As edges between elements of S_1 are removed in F^*, this cycle is not present in F^*. The dependency between $\mathbf{a}$ and $\mathbf{b}$ is now accounted for by the variable $\mathbf{s}_1$, which acts as a common cause for $\mathbf{a}$ and $\mathbf{b}$. Now consider the SCC $S_3 = \{\mathbf{e}, \mathbf{f}, \mathbf{g}\}$, which is transformed similarly, but in addition, the attack of the parents $\mathbf{b}$ and $\mathbf{d}$ of S_3 on $\mathbf{e}$ is replaced with edges from $\mathbf{b}$ and $\mathbf{d}$ to $\mathbf{s}_3$. This is done so that the dependence of *all* elements of S_3 on $\mathbf{b}$ and $\mathbf{d}$—not just $\mathbf{e}$ but also $\mathbf{f}$ and $\mathbf{g}$—is still accounted for. Note that the addition of the variable $\mathbf{s}_4$ for the singleton SCC S_4 is not actually needed, but treating all SCCs the same simplifies the definition of d-graph.

6.1 Soundness of the D-graph Approach

Given a graph G, the d-graph G^* is a DAG that contains all variables present in G. Thus, we can use d-separation in G^* to derive independence statements about a belief state over G. If the dependency model of this belief state satisfies the SCC Markov condition with respect to G, then these statements are true.

Theorem 3. *Let T be a belief state over $G = (\mathbf{X}, \rightarrow)$. If I_T satisfies the SCC-Markov condition with respect to G then for all disjoint sets $A, B, C \subseteq \mathbf{X}$ we have that $D_{G^*}(A, B \mid C)$ implies $I_T(A, B \mid C)$.*

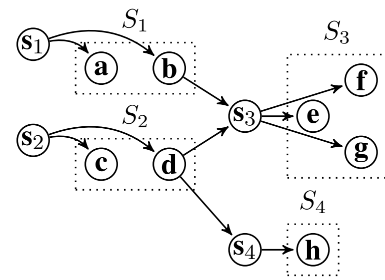


Figure 4: The d-graph for the AF from Figure 2

Proof. Let T be a belief state over the graph $G = (\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \rightarrow)$ with SCCs $\{S_1, \ldots, S_m\}$. Suppose I_T satisfies the SCC Markov condition with respect to G. Then let $(G, \{\Phi_{S_1}, \ldots, \Phi_{S_m}\})$ be the causal theory that represents T, the existence of which follows from Lemma 1. Let $G^* = (\{\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{s}_1, \ldots, \mathbf{s}_m\}, \rightarrow')$ be the d-graph of G where the domain of $\mathbf{s}_i$ is defined by $Dom(\mathbf{s}_i) = \mathcal{V}(S_i)$ (i.e., the values of $\mathbf{s}_i$ are the valuations of S_i). Now define the causal theory $(G^*, \{\Phi_{\mathbf{x}_1}, \ldots, \Phi_{\mathbf{x}_n}, \Phi_{\mathbf{s}_1}, \ldots, \Phi_{\mathbf{s}_m}\})$ by

$$\forall V \in \mathcal{V}(\{\mathbf{s}_i\}), V \in \Phi_{\mathbf{s}_i}(V') \text{ iff } V(\mathbf{s}_i) \in \Phi_{S_i}(V'),$$

$$\Phi_{\mathbf{x}_i}(V) = V(\mathbf{s}_j)(\mathbf{x}_i), \text{where } j \text{ is s.t. } \mathbf{x}_i \in S_j,$$

and let T^* be the theory it represents. Then:

1. the definition of C implies that $T{\downarrow}\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} = T^*$. Hence for all disjoint sets $A, B, C \subseteq \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ we have $I_{T^*}(A, B \mid C)$ if and only if $I_T(A, B \mid C)$.
2. Lemma 1 implies that I_{T^*} satisfies the SCC Markov condition with respect to G^*. Since G^* is acyclic, this implies, using Proposition 2, that I_{T^*} satisfies the local Markov condition with respect to G^*.

These two facts imply, using Theorem 1, that for all disjoint sets $A, B, C \subseteq \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ we have that $D_{G^*}(A, B \mid C)$ implies $I_T(A, B \mid C)$. □

A corollary of this theorem is:

Corollary 1. *Let σ be an SCC-Markovian semantics and let $F = (\mathbf{A}, \Rightarrow)$ be an AF. Then for all disjoint sets $A, B, C \subseteq \mathbf{A}$ we have that $D_{F^*}(A, B \mid C)$ implies $I^\sigma_F(A, B \mid C)$.*

In the following example we show that the independencies that were shown to hold in Example 2 can all be derived using the d-graph approach.

Example 5. *Let F be the AF shown earlier in Figure 2. The d-graph F^* of F is shown in Figure 4. In Example 2 we listed a number of (non-)independencies that hold in I^{pr}_F. In this example we show that all these independencies are derivable using d-separation in F^*, while none of the non-independencies are derivable.*

In the d-graph F^* we have that:

1. *$\mathbf{a}$ and $\mathbf{e}$ are not d-separated by the empty set because the empty set does not block the trail $\mathbf{a} \leftarrow \mathbf{s}_1 \rightarrow \mathbf{b} \rightarrow \mathbf{s}_3 \rightarrow \mathbf{e}$. This trail is blocked by $\mathbf{b}$, however, which is due to the chain $\mathbf{s}_1 \rightarrow \mathbf{b} \rightarrow \mathbf{s}_3$. Therefore $\mathbf{a}$ and $\mathbf{e}$ are d-separated by $\mathbf{b}$. Thus, we have:*

$$\neg D_{F^*}(\mathbf{a}, \mathbf{e} \mid \emptyset) \qquad D_{F^*}(\mathbf{a}, \mathbf{e} \mid \mathbf{b})$$

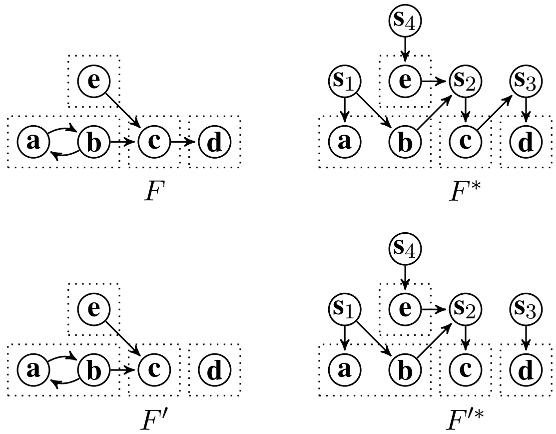


Figure 5: Independence of $\mathbf{b}$ and $\mathbf{d}$ derivable after pruning

2. $\mathbf{e}$ *and* $\mathbf{h}$ *are not d-separated by the empty set because the empty set does not block the trail* $\mathbf{e} \leftarrow \mathbf{s}_3 \leftarrow \mathbf{d} \rightarrow \mathbf{s}_4 \rightarrow h$. *This trail is blocked by* $\mathbf{d}$, *however, due to the fork* $\mathbf{s}_3 \leftarrow \mathbf{d} \rightarrow \mathbf{s}_4$. *Therefore* $\mathbf{e}$ *and* $\mathbf{h}$ *are d-separated by* $\mathbf{d}$. *Thus, we have:*

$$\neg D_{F^*}(\mathbf{e}, \mathbf{h} \mid \emptyset) \qquad D_{F^*}(\mathbf{e}, \mathbf{h} \mid \mathbf{d})$$

3. *The sets* $\{\mathbf{a}, \mathbf{b}\}$ *and* $\{\mathbf{c}, \mathbf{d}\}$ *are d-separated by the empty set. This is because the empty set blocks all trails between the two sets. However, they are not d-separated by* $\mathbf{e}$, *because of the trail* $\mathbf{b} \rightarrow \mathbf{s}_3 \leftarrow \mathbf{d}$ *and the fact that* $\mathbf{e}$ *is a descendant of* $\mathbf{s}_3$. *They are similarly not d-separated by* $\mathbf{f}$ *or* $\mathbf{g}$, *which are also descendants of* $\mathbf{s}_3$. *Thus, we have:*

$$D_{F^*}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \emptyset) \qquad \neg D_{F^*}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \mathbf{f})$$
$$\neg D_{F^*}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \mathbf{e}) \qquad \neg D_{F^*}(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{c}, \mathbf{d}\} \mid \mathbf{g})$$

4. $\mathbf{a}$ *and* $\mathbf{h}$ *are d-separated by the empty set. This is because the empty set blocks the trail* $\mathbf{a} \leftarrow \mathbf{s}_1 \rightarrow \mathbf{b} \rightarrow \mathbf{s}_3 \leftarrow \mathbf{d} \rightarrow \mathbf{s}_4 \rightarrow \mathbf{h}$ *due to the collider* $\mathbf{b} \rightarrow \mathbf{s}_3 \leftarrow \mathbf{d}$. *This trail is no longer blocked if we observe* $\mathbf{e}$, *which is a descendant of* $\mathbf{s}_3$. *If we additionally observe* $\mathbf{d}$ *then the trail is blocked again due to the fork* $\mathbf{s}_3 \leftarrow \mathbf{d} \rightarrow \mathbf{s}_4$. *Thus, we have:*

$$D_{F^*}(\mathbf{a}, \mathbf{h} \mid \emptyset) \quad \neg D_{F^*}(\mathbf{a}, \mathbf{h} \mid \mathbf{e}) \quad D_{F^*}(\mathbf{a}, \mathbf{h} \mid \{\mathbf{d}, \mathbf{e}\})$$

6.2 (In)completeness of the D-graph Approach

In the previous section we proved the soundness of the d-graph approach: if σ is SCC Markovian then $D_{F^*}(A, B \mid C)$ implies $I^\sigma_F(A, B \mid C)$. Example 5 also shows that the approach is fairly complete; in all cases considered there, we have that $I^\sigma_F(A, B \mid C)$ implies $D_{F^*}(A, B \mid C)$. The approach is not fully complete, however. A number of cases can be identified where independencies that hold under an SCC Markovian semantics such as the complete or preferred semantics cannot be derived using the d-graph approach.

One case concerns independencies that hold because an effect gets cancelled out due to mediating arguments being rejected. This is demonstrated by the AF F and d-graph F^* shown in Figure 5. Here $\mathbf{e}$ is always accepted and attacks $\mathbf{c}$. The indirect effect of $\mathbf{b}$ on $\mathbf{d}$, which is mediated by $\mathbf{c}$, is therefore cancelled out, making $\mathbf{b}$ and $\mathbf{d}$ independent. We cannot, however, derive independence of $\mathbf{b}$ and $\mathbf{d}$ using d-separation in the d-graph F^*, which still contains a path from $\mathbf{b}$ to $\mathbf{d}$. Fortunately, this case is easily dealt with by pruning the AF before transforming it into a d-graph. In this pruning step, we remove attacks originating from arguments that are attacked by unattacked arguments, repeating this operation until no more attacks are removed. The AF F' shown in Figure 5 shows the result of applying this pruning step to F. As $\mathbf{b}$ and $\mathbf{d}$ are no longer connected in the resulting d-graph F'^*, their independence can now be derived. This solution applies to any semantics under which the pruning step described here does not change the set of labellings, such as the complete and preferred semantics. Note that the attacks removed in the pruning step are exactly the attacks coming from arguments that are labelled `O` in the grounded labelling of the AF. This pruning step therefore requires polynomial time.

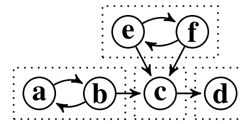


Figure 6: Independence of $\mathbf{b}$ and $\mathbf{d}$ not derivable

We cannot always easily detect effects that are cancelled out due to mediating arguments being rejected. Consider, for example, the AF F shown in Figure 6. Under the preferred semantics, $\mathbf{c}$ is always rejected as it is attacked by $\mathbf{e}$ and $\mathbf{f}$, of which one is always accepted. Thus, like the example above, the effect of $\mathbf{b}$ on $\mathbf{d}$ is cancelled out, but we cannot derive independence of $\mathbf{b}$ and $\mathbf{d}$ using d-separation in F^*. Unfortunately, a single syntactic pruning criterion to solve all instances exhibiting this problem does not seem to exist.

A final case of incompleteness concerns observations of arguments that only attack arguments within the same SCC. This is demonstrated by the AF shown in Figure 7. Here, the arguments $\mathbf{c}$ and $\mathbf{d}$ belong to the same SCC and $\mathbf{c}$ attacks only $\mathbf{d}$. This results in a d-graph in which $\mathbf{c}$ has just one parent and no children, which means that $\mathbf{c}$ cannot block any trail. As a consequence, the independence of $\mathbf{b}$ and $\mathbf{e}$ given $\mathbf{c}$, which holds under all of the common semantics, cannot be derived. The root of the problem here is that independence of $\mathbf{b}$ and $\mathbf{e}$ given $\mathbf{c}$ is not in fact implied by SCC Markovianness. Indeed, it is possible to define a semantics that is SCC Markovian under which $\mathbf{b}$ and $\mathbf{e}$ are not independent given $\mathbf{c}$ (e.g., evaluate $\{\mathbf{a}, \mathbf{b}\}$ according to the preferred semantics, let $\mathbf{c}$ always be rejected and assign to $\mathbf{d}$ the same label as $\mathbf{b}$). To uncover these independencies we therefore need to go beyond the SCC Markov condition and consider stronger decomposability principles that account for independencies on a more fine-grained level. We plan to investigate this in future work.

Let us point out, however, that if independence information is used for the purpose of speeding up inference then some degree of incompleteness is not critical. Incompleteness just means that we sometimes need to fall back on a

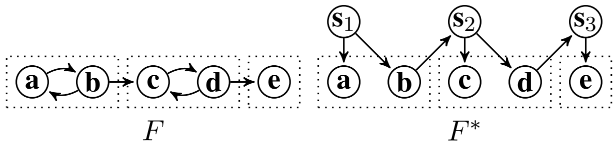


Figure 7: Independence of $\mathbf{b}$ and $\mathbf{e}$ given $\mathbf{c}$ not derivable

(typically more expensive) procedure that does not rely on independence, even though a (typically cheaper) procedure that does rely on independence would also have worked. However, any procedure that does not rely on independence should also produce correct results when independence does hold. Soundness, on the other hand, is critical, because assuming independence when independence does not hold may lead to incorrect results.

6.3 Complexity of the D-Graph Approach

Determining independence using the d-graph approach is computationally cheap. The first step, which is the d-graph transformation, amounts to identifying the SCCs of the graph, which is known to take linear time in the number of nodes and edges (e.g., using *Kosaraju's* algorithm). The extra pruning step described in Section 6.2 adds polynomial time. However, this needs to be done only once, because a d-graph can be reused for multiple d-separation tests. A d-separation test itself is also known to take linear time (Darwiche 2009). Taking everything into account, determining independence is therefore tractable. By contrast, many inference problems in abstract argumentation are intractable (Kröll, Pichler, and Woltran 2017; Dvořák and Dunne 2018). Many problems whose runtime can be reduced by using independence information may therefore benefit from our approach.

7 Discussion

A notion of *relevance* in abstract argumentation was studied by (Liao and Huang 2013). The definition is simple: an argument $\mathbf{x}$ is relevant to an argument $\mathbf{y}$ if and only if a directed path from $\mathbf{x}$ to $\mathbf{y}$ exists. This notion of relevance applies to semantics that satisfy the Directionality criterion (Baroni, Giacomin, and Guida 2005):

Definition 16. *A semantics σ satisfies* Directionality *if and only if, for every AF $F = (\mathbf{A}, \Rightarrow)$, and every unattacked set U of F, we have that*

$$\mathcal{L}_\sigma(F{\downarrow}U) = \mathcal{L}_\sigma(F){\downarrow}U,$$

where a set $U \subseteq \mathbf{A}$ is an unattacked set of F if and only if no $\mathbf{x} \in \mathbf{A} \setminus U$ attacks any $\mathbf{y} \in U$.

They show that, under any directional semantics, one may remove arguments not relevant to a set B if one is interested in computing only the status of B, which can significantly reduce runtime. The same principle is used in (Liao, Jin, and Koons 2011) to reduce runtime of recomputing the extensions of an AF when the AF is changed. We can define *irrelevance* as the complement of relevance. Irrelevance of this kind does not imply independence, however, since absence of a directed path from $\mathbf{x}$ to $\mathbf{y}$ does not imply independence of $\mathbf{x}$ and $\mathbf{y}$ (see, e. g., the arguments $\mathbf{e}$ and $\mathbf{h}$ in Example 5). On the other hand, unconditional d-separation of $\mathbf{x}$ and $\mathbf{y}$ implies absence of a directed path—in either direction—between $\mathbf{x}$ and $\mathbf{y}$ and thus implies irrelevance in both directions. This notion of irrelevance is, however, unconditional, so it cannot be compared to independence in the conditional case. Furthermore, Directionality neither implies nor is implied by SCC Markovianness. Irrelevance and independence are therefore based on different underlying principles. An example of a directional but non-SCC-Markovian semantics is the semantics that labels arguments either all `I` or all `O`, except for self-attacking arguments, which are labelled `O`. An example of an SCC Markovian but non-directional semantics is the semantics under which, for any single-argument AF $(\{x\}, \Rightarrow)$, x is labelled `O`, while every other AF admits every possible labelling. Nevertheless, among the semantics considered in the literature, those that satisfy SCC Markovianness appear to be precisely those that satisfy Directionality. In future work we plan to investigate the relationship between SCC Markovianness and Directionality, as well as the relationship between independence and irrelevance, in more detail.

The SCC Decomposability principle is related to the SCC recursiveness principle (Baroni, Giacomin, and Guida 2005). SCC recursiveness, in short, means that the semantics is definable in terms of a *base function* that operates—like the local function in Definition 11—on each SCC separately, but also recursively on smaller SCCs that form after removing arguments that are defeated. It is not hard to prove that SCC recursiveness implies SCC decomposability. Any semantics that is universal and SCC recursive is therefore SCC Markovian. Examples are the *CF2* and *Stage2* semantics (Dvorák and Gaggl 2016).

The probabilistic analogue of the SCC decomposability principle for probabilistic abstract argumentation was considered in (Rienstra et al. 2018). A similar ranking-based analogue was also investigated in (Rienstra and Thimm 2018). Use of d-separation to determine probabilistic or ranking-based conditional independence was not considered in these papers, although the d-graph method used in this paper can be applied in these settings too.

8 Summary and Conclusion

In this paper we defined a notion of conditional independence in abstract argumentation. We showed that the d-separation criterion can be used to derive independencies from the structure of what we call the *d-graph* of an AF, and that these independencies hold under any semantics that is SCC Markovian.

While many inference tasks in abstract argumentation are intractable, testing for independence using our approach is tractable. Thus, many inference tasks can potentially benefit from using independence information. In the introduction we already discussed some examples. In future work we plan to study the practical use of independence in argumentation in more detail.

References

Baroni, P.; Boella, G.; Cerutti, F.; Giacomin, M.; van der Torre, L.; and Villata, S. 2014. On the input/output behavior of argumentation frameworks. *Artificial Intelligence* 217:144–197.

Baroni, P.; Giacomin, M.; and Guida, G. 2005. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence* 168(1-2):162–210.

Borg, A., and Straßer, C. 2018. Relevance in structured argumentation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*.

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

Darwiche, A., and Pearl, J. 1994. Symbolic causal networks. In Hayes-Roth, B., and Korf, R. E., eds., *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1*, 238–244. AAAI Press / The MIT Press.

Darwiche, A. 1997. A logical notion of conditional independence: properties and applications. *Artificial Intelligence* 97(1-2):45–82.

Darwiche, A. 2009. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press.

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

Dvorák, W., and Gaggl, S. A. 2016. Stage semantics and the scc-recursive schema for argumentation semantics. *J. Log. Comput.* 26(4):1149–1202.

Dvořák, W., and Dunne, P. E. 2018. Computational problems in formal argumentation and their complexity. In Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds., *Handbook of Formal Argumentation*. College Publications. chapter 14.

Koller, D., and Halpern, J. Y. 1996. Irrelevance and conditioning in first-order probabilistic logic. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, 569–576. AAAI Press.

Kourousias, G., and Makinson, D. C. 2007. Parallel interpolation, splitting, and relevance in belief change. *Journal of Symbolic Logic* 72:994–1002.

Kröll, M.; Pichler, R.; and Woltran, S. 2017. On the complexity of enumerating the extensions of abstract argumentation frameworks. In Sierra, C., ed., *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1145–1152. ijcai.org.

Liao, B., and Huang, H. 2013. Partial semantics of argumentation: basic properties and empirical. *J. Log. Comput.* 23(3):541–562.

Liao, B. S.; Jin, L.; and Koons, R. C. 2011. Dynamics of argumentation systems: A division-based method. *Artificial Intelligence* 175(11):1790–1814.

Pearl, J. 1989. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann.

Pearl, J. 2000. *Causality: models, reasoning and inference*, volume 29. Cambridge University Press.

Rienstra, T., and Thimm, M. 2018. Ranking functions over labelings. In Modgil, S.; Budzynska, K.; and Lawrence, J., eds., *Computational Models of Argument - Proceedings of COMMA 2018, Warsaw, Poland, 12-14 September 2018*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, 393–404. IOS Press.

Rienstra, T.; Thimm, M.; Liao, B.; and van der Torre, L. W. N. 2018. Probabilistic abstract argumentation based on SCC decomposability. In Thielscher, M.; Toni, F.; and Wolter, F., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018*, 168–177. AAAI Press.

Verma, T., and Pearl, J. 1990. Causal networks: Semantics and expressiveness. In *Machine intelligence and pattern recognition*, volume 9. Elsevier. 69–76.