Measuring Inconsistency in Argument Graphs

Anthony Hunter

Department of Computer Science
University College London, London, UK
anthony.hunter@ucl.ac.uk

Abstract

There have been a number of developments in measuring inconsistency in logic-based representations of knowledge. In constrast, the development of inconsistency measures for computational models of argument has been limited. To address this shortcoming, this paper provides a general framework for measuring inconsistency in abstract argumentation, together with some proposals for specific measures, and a consideration of measuring inconsistency in logic-based instantiations of argument graphs, including a review of some existing proposals and a consideration of how existing logic-based measures of inconsistency can be applied.

1 Introduction

Argumentation is an important cognitive ability for handling conflicting and incomplete information such as beliefs, assumptions, opinions, and goals. When we are faced with a situation where we find that our information is incomplete or inconsistent, we often resort to the use of arguments for and against a given position in order to make sense of the situation. Furthermore, when we interact with other people we often exchange arguments to reach a final agreement and/or to defend and promote an individual position.

In recent years, there has been substantial interest in the development of computational models of argument for capturing aspects of this cognitive ability (for reviews see [BCD07, BH08, RS09]). This has led to the development of a number of directions including: (1) abstract argument models where arguments are atomic, and the emphasis is on the relationships between arguments; (2) logic-based (or structured) argument models where the emphasis is on the logical structure of the premises and claim of the arguments, and the logical definition of relations between arguments; and (3) dialogical argument models where the emphasis is on the protocols (i.e. allowed and obligatory moves that can be taken at each step of the dialogue) and strategies (i.e. mechanisms

used by each participant to make the best choice of move at each step of the dialogue).

At the core of computational models of argument is the ability to represent and reason with inconsistency. So it is perhaps surprising that relatively little consideration has been given to measuring inconsistency in these models, particularly given the number of developments in measuring inconsistency in logic-based knowledgebases (see for example [Kni01, HK04, DRMO10, HK10, GH11b, MLJ12, JMR14, Bes14, Thi16]).

A couple of exceptions are the consideration of the degree of undercut between an argument and counterargument [BH05, BH08], and measuring inconsistency through argumentation [Rad15]. Note, the approach of weighted argumentation frameworks [DHM$^+$11] is not a measure of inconsistency as the approach assumes extra information (weights) to label each arc, and an inconsistency budget that allows arcs that sum to no more than the budget to be ignored.

There are a number of reasons why it is useful to investigate the measurement of inconsistency in argumentation: (1) to better characterize the nature of inconsistency in argumentation; (2) to analyse the inconsistency arising in specific argumentation situations; and (3) to direct the resolution of inconsistency as arising in argumentation. We will consider contributions to these three areas during the course of this chapter.

Given the central role of argument graphs (where each node is an argument and each arc denotes one argument attacking another) in modelling argumentation, we will consider the inconsistency of an argument graph. This is useful if we want to assess the overall conflict that is manifested by an argument graph, and we want to focus on actions that may allow us to decrease the graph inconsistency.

Consider for example some security analysts who are analyzing some conflicting reports concerning a foreign country that may be descending into civil war. These analysts may enter into a process as follows: (1) they collect relevant information concerning the political and security situation in the country; (2) they construct arguments from this information that draw tentative hypotheses about the situation in the country; (3) they compose these arguments into an argument graph; (4) they measure the inconsistency of the argument graph; (5) they use the measure of inconsistency to identify information requirements (i.e. queries to ascertain whether a particular argument should be accepted or rejected) that would result in commitments being made for some of these arguments; (6) they seek the answers to these queries; (7) they use these commitments to reduce the overall inconsistency of the graph; and (8) they terminate this process when sufficient commitments have been made so as to reduce the inconsistency to a sufficiently low level.

This kind of process may be of relevance to security analysts to augment recent proposals for argument-based security analysis technology such as by

Toniolo et al [TNE$^+$15]. Furthermore, this kind of process may be replicated in roles such as business intelligence analysis, policy planning, political planning, and science research.

Before we proceed, we need to consider why we are using the term inconsistency, and what we mean by it. When we use argumentation, we are normally considering situations where there exist both arguments and counterarguments. Suppose we have just two arguments A and B where B attacks A. Here, we regard this as an inconsistent situation because accepting both argument would not be conflict-free. In other words, there is an incompatibility between the arguments. One could argue that if we use the dialectical semantics proposed by Dung (as we will review in the next section), we obviate this problem since we would accept B and not accept A, and so in a sense, the inconsistency is resolved. But that does not mean that there is no inconsistency. Indeed, if A has no role, then why even present A in the argument graph? Yet we do present A because part of the role of argumentation is to consider the conflicts that arise, and then make sense of those conflicts by choosing for instance which argument to accept. Choosing which arguments to accept may be by using dialectical semantics. But that is only one course of action. For instance, we may have reasons to doubt B, and so it may then be reasonable to seek further information about B to either confirm it or to reject it (as indicated by the intelligence analysis scenario discussed above). So in general, we see a key part of argumentation is to identify inconsistencies, and then to have mechanisms for dealing with those inconsistencies. Hence, the purpose of this paper is to consider how we might measure these inconsistencies.

We proceed as follows: (Section 2) We review the basic definitions of abstract argumentation, considering both extension-based and label-based approaches; (Section 3) We investigate a general framework for measuring inconsistency in abstract argumentation, together with some proposals for specific measures; (Section 4) We review deductive argumentation for instantiating abstract argument graphs, we review an existing proposal for measuring inconsistency in deductive argumentation called degree of undercut, and we investigate a new approach that harnesses existing logic-based measures; (Section 5) We consider how we can use measures of inconsistency to direct the resolution of inconsistency in argumentation; (Section 6) We conclude with a discussion of the proposals in the chapter and of future work.

2 Review of abstract argumentation

Our framework builds on more general developments in the area of computational models of argument. These models aim to reflect how human argumentation uses conflicting information to construct and analyze arguments.

There is a number of frameworks for computational models of argumen-

3

tation. They incorporate a formal representation of individual arguments and techniques for comparing conflicting arguments (for reviews see [BCD07, BH08, RS09, ABG$^+$17, BGGv18]). By basing our framework on these models, we can harness theory and tools as the basis of our solution.

2.1 Extension-based semantics

We start with a brief review of abstract argumentation as proposed by Dung [Dun95]. In this approach, each argument is treated as an atom, and so no internal structure of the argument needs to be identified.

Definition 1. *An* **argument graph** *is a pair* $G = (\mathcal{A}, \mathcal{R})$ *where* $\mathcal{A}$ *is a set and* $\mathcal{R}$ *is a binary relation over* $\mathcal{A}$ *(in symbols,* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ *). Let* $\mathsf{Nodes}(G)$ *be the set of nodes in* G *and let* $\mathsf{Arcs}(G) \subseteq \mathsf{Nodes}(G) \times \mathsf{Nodes}(G)$ *be the set of arcs in* G.

So an argument graph is a directed graph. Each element $A \in \mathcal{A}$ is called an **argument** and $(A_i, A_j) \in \mathcal{R}$ means that A_i **attacks** A_j (accordingly, A_i is said to be an **attacker** of A_j). So A_i is a **counterargument** for A_j when $(A_i, A_j) \in \mathcal{R}$ holds.

Example 1. *Consider arguments* $A_1 =$ *"Patient has hypertension so prescribe diuretics",* $A_2 =$ *"Patient has hypertension so prescribe beta-blockers", and* $A_3 =$ *"Patient has emphysema which is a contraindication for beta-blockers". Here, we assume that* A_1 *and* A_2 *attack each other because we should only give one treatment and so giving one precludes the other, and we assume that* A_3 *attacks* A_2 *because it provides a counterargument to* A_2*. Hence, we get the following abstract argument graph.*
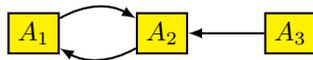
Arguments can work together as a coalition by attacking other arguments and by defending their members from attack as follows.

Definition 2. *Let* $S \subseteq \mathcal{A}$ *be a set of arguments.*

- *S **attacks** $A_j \in \mathcal{A}$ iff there is an argument $A_i \in S$ such that A_i attacks A_j.*

- *S **defends** $A_i \in S$ iff for each argument $A_j \in \mathcal{A}$, if A_j attacks A_i then S attacks A_j.*

The following gives a requirement that should hold for a coalition of arguments to make sense. If it holds, it means that the arguments in the set offer a consistent view on the topic of the argument graph.

Definition 3. *A set $S \subseteq \mathcal{A}$ of arguments is* **conflict-free** *iff there are no arguments A_i and A_j in S such that A_i attacks A_j.*

Now, we consider how we can find an acceptable set of arguments from an abstract argument graph. The simplest case of arguments that can be accepted is as follows.

Definition 4. *A set $S \subseteq \mathcal{A}$ of arguments is* **admissible** *iff S is conflict-free and defends all its arguments.*

The intuition here is that for a set of arguments to be accepted, we require that, if any one of them is challenged by a counterargument, then they offer grounds to challenge, in turn, the counterargument. There always exists at least one admissible set: The empty set is always admissible.

Clearly, the notion of admissible sets of arguments is the minimum requirement for a set of arguments to be accepted. We will focus on the following classes of acceptable arguments.

Definition 5. *Let Γ be a conflict-free set of arguments, and let* Defended $: \wp(\mathcal{A}) \mapsto \wp(\mathcal{A})$ *be a function such that* Defended$(\Gamma) = \{A \mid \Gamma \text{ defends } A\}$.

1. *Γ is a* **complete extension** *iff $\Gamma = $ Defended(Γ)*

2. *Γ is a* **grounded extension** *iff it is the minimal (w.r.t. set inclusion) complete extension.*

3. *Γ is a* **preferred extension** *iff it is a maximal (w.r.t. set inclusion) complete extension.*

4. *Γ is a* **stable extension** *iff it is a preferred extension that attacks every argument that is not in the extension.*

The grounded extension is always unique, whereas there may be multiple preferred extensions. We illustrate these definitions with the following examples. As can be seen from the examples, the grounded extension provides a skeptical view on which arguments can be accepted, whereas each preferred extension takes a credulous view on which arguments can be accepted.

Example 2. *Continuing Example 1, there is only one complete set, and so this is both grounded and preferred. Note, $\{A_1, A_2\}$, $\{A_2, A_3\}$, and $\{A_1, A_2, A_3\}$ are not conflict-free subsets. Only the conflict-free subsets are given in the table.*

	Conflict-free	Admiss-ble	Complete	Ground-ed	Preferr-ed	Stable
$\{\}$	✓	✓	×	×	×	×
$\{A_1\}$	✓	✓	×	×	×	×
$\{A_2\}$	✓	×	×	×	×	×
$\{A_3\}$	✓	✓	×	×	×	×
$\{A_1, A_3\}$	✓	✓	✓	✓	✓	✓

Example 3. *Consider the following argument graph.*

For this, there are two preferred sets, neither of which is grounded. Note $\{A_4, A_5\}$ is not conflict-free. Only the conflict-free subsets are given in the table.

	Conflict -free	Admissible	Complete	Grounded	Preferred	Stable
$\{\}$	✓	✓	✓	✓	×	×
$\{A_4\}$	✓	✓	✓	×	✓	✓
$\{A_5\}$	✓	✓	✓	×	✓	✓

The formalization we have reviewed in this section is abstract because both the nature of the arguments and the nature of the attack relation are ignored. In particular, the internal (logical) structure of each of the arguments is not made explicit. Nevertheless, Dung's proposal for abstract argumentation is ideal for clearly representing arguments and counterarguments, and for intuitively determining which arguments should be accepted (depending on whether we want to take a credulous or skeptical perspective).

Given an argument graph, let $\mathsf{Extensions}_\sigma(G)$ denote the set of extensions according to σ where $\sigma = \mathsf{co}$ denotes the complete extensions, $\sigma = \mathsf{pr}$ denotes the preferred extensions, $\sigma = \mathsf{gr}$ denotes the grounded extensions, and $\sigma = \mathsf{st}$ denotes the stable extensions.

2.2 Labelling-based semantics

We now review an alternative way of defining semantics for abstract argumentation proposed by Caminada and Gabbay [CG09]. A labelling L is a function $L : \mathsf{Nodes}(G) \to \{\mathsf{in}, \mathsf{out}, \mathsf{undec}\}$ that assigns to each argument $A \in \mathsf{Nodes}(G)$ either the value in, meaning that the argument is accepted, out, meaning that the argument is not accepted, or undec, meaning that the status of the argument is undecided. Let $\mathsf{in}(L) = \{A \mid L(A) = \mathsf{in}\}$ and $\mathsf{out}(L)$ resp. $\mathsf{undec}(L)$ be defined analogously. The set $\mathsf{in}(L)$ for a labelling L is also called an *extension*. A labelling L is called *conflict-free* if for no $A, B \in \mathsf{in}(L)$ we have that $(A, B) \in \mathsf{Arcs}(G)$.

Definition 6. *A labelling L is called* **admissible** *if and only if for all arguments $A \in \mathsf{Nodes}(G)$*

1. *if $L(A) = \mathsf{out}$ then there is $B \in \mathsf{Nodes}(G)$ with $L(B) = \mathsf{in}$ and $(B, A) \in \mathsf{Arcs}(G)$, and*

6

2. *if $L(A) = $ in then $L(B) = $ out for all $B \in$ Nodes(G) with $(B, A) \in$ Arcs(G),*

and it is called **complete** *if, additionally, it satisfies*

3. *if $L(A) = $ undec then there is no $B \in$ Nodes(G) with $(B, A) \in$ Arcs(G) and $L(B) = $ in and there is a $B' \in$ Nodes(G) with $(B', A) \in$ Arcs(G) and $L(B') \neq$ out.*

The intuition behind admissibility is that an argument can only be accepted if there are no attackers that are accepted and if an argument is not accepted then there have to be some reasonable grounds. The idea behind the completeness property is that the status of an argument is only undec if it cannot be classified as in or out. Different types of classical semantics can be obtained by imposing further constraints.

Definition 7. *Let G be an argument graph, let $L :$ Nodes$(G) \rightarrow \{$in, out, undec$\}$ be a complete labelling, and for all statements, minimality/maximality is with respect to set inclusion.*

- *L is* **grounded** *if and only if* in(L) *is minimal.*

- *L is* **preferred** *if and only if* in(L) *is maximal.*

- *L is* **stable** *if and only if* undec$(L) = \emptyset$.

Example 4. *Continuing Example 2, there is one complete labelling.*

	A_1	A_2	A_3	Type
L_1	in	out	in	*grounded, preferred, stable*

Example 5. *Continuing Example 3, there are three complete labellings.*

	A_4	A_5	Type
L_1	undec	undec	*grounded*
L_2	in	out	*preferred, stable*
L_3	out	in	*preferred, stable*

The extension-based semantics and labelling-based semantics are equivalent. So for instance, for the grounded extension Γ for argument graph G, and the grounded labelling G, we have $\Gamma = $ in(L). Similarly, each preferred extension (respectively stable) is represented by a preferred (respectively stable) labelling.

2.3 Subsidiary definitions

We now consider some further simple definitions that we will use as subsidiary functions for our measures of inconsistency for abstract argument graphs.

Definition 8. *For a graph G, and an argument A, the indegree and outdegree functions are defined as follows.*

- $\mathsf{Indegree}(G, A) = |\{(B, A) \mid (B, A) \in \mathsf{Arcs}(G)\}|$

- $\mathsf{Outdegree}(G, A) = |\{(A, B) \mid (A, B) \in \mathsf{Arcs}(G)\}|$

**Example 6.** *For the graph below, we get the following indegree and outdegree values.*

$$
\begin{array}{ll}
\mathsf{Indegree}(G, A_1) = 0 & \mathsf{Outdegree}(G, A_1) = 2 \\
\mathsf{Indegree}(G, A_2) = 3 & \mathsf{Outdegree}(G, A_2) = 2 \\
\mathsf{Indegree}(G, A_3) = 2 & \mathsf{Outdegree}(G, A_3) = 3 \\
\mathsf{Indegree}(G, A_4) = 2 & \mathsf{Outdegree}(G, A_4) = 0
\end{array}
$$
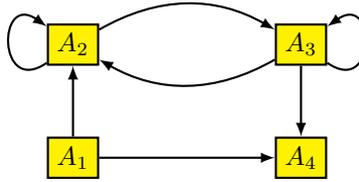
Definition 9. *For graphs G_1 and G_2, G_1 is a **subgraph** of G_2, denoted $G_1 \sqsubseteq G_2$, iff*
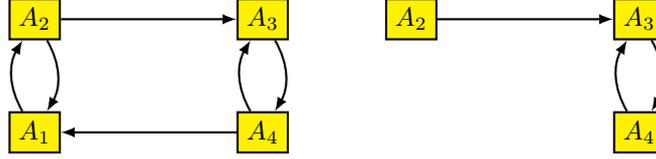
$\mathsf{Nodes}(G_1) \subseteq \mathsf{Nodes}(G_2)$ *and* $\mathsf{Arcs}(G_1) \subseteq (\mathsf{Arcs}(G_2) \cap (\mathsf{Nodes}(G_1) \times \mathsf{Nodes}(G_1)))$

Example 7. *For the following graphs G_1 (left) and G_2 (right), $G_2 \sqsubseteq G_1$ holds.*
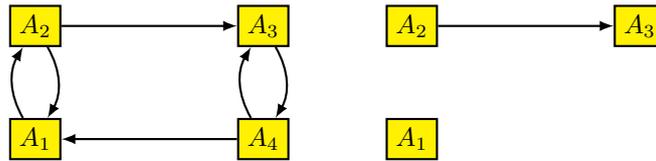
Definition 10. *For graph G, and a set of arguments $X \subseteq \mathsf{Nodes}(G)$, the **induced graph** is a graph G' such that $G' \sqsubseteq G$ and $\mathsf{Nodes}(G') = X$ and $\mathsf{Arcs}(G') = \mathsf{Arcs}(G) \cap (X \times X)$. Let $\mathsf{Induce}(G, X)$ be the function that returns the induced graph for G and X.*
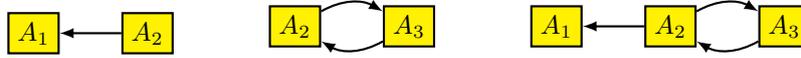
Example 8. *For the graph G (left), and $X = \{A_2, A_3, A_4\}$, the induced graph is $\mathsf{Induce}(G, X)$ (right)*

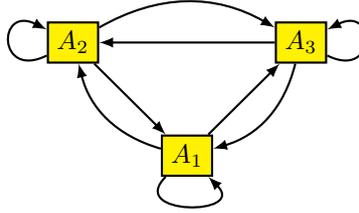**Definition 11.** *For graphs G_1 and G_2, the* **graph union** *of G_1 and G_2, denoted $G_1 + G_2$, is*

$$(\mathsf{Nodes}(G_1) \cup \mathsf{Nodes}(G_2)), (\mathsf{Arcs}(G_1) \cup \mathsf{Arcs}(G_2))$$

Example 9. *For the graph G_1 (left), G_2 (middle), and $G_1 + G_2$ (right).*

Definition 12. *A graph G is* **complete** *iff for all $A, B \in \mathsf{Nodes}(G)$, $(A, B) \in \mathsf{Arcs}(G)$.*

Example 10. *The following is a complete argument graph with three nodes.*

In the following definition, we define a cycle as a subset of nodes in the graph for which there is a circular path involving all these nodes.

Definition 13. *For an arc $E = (A, B)$, let $\mathsf{Source}(E) = A$ and $\mathsf{Target}(E) = B$. For a graph G, a* **path** *in G is a sequence of arcs $E_1, ... E_n$ such that for each E_i, where $1 \leq i < n$, $\mathsf{Target}(E_i) = \mathsf{Source}(E_{i+1})$, and for each E_i, E_j, where $1 \leq i, j \leq n$, if $i \neq j$, then $E_i \neq E_j$. A* **cyclical path** *in G is a path $E_1, ... E_n$ such that $\mathsf{Target}(E_n) = \mathsf{Source}(E_1)$. A* **cycle** *is the set of nodes appearing in the arcs in a cyclical path (i.e. if $E_1, \ldots, E_n$ is a cyclical path, then $\mathsf{Cycle}(\{E_1, \ldots, E_n\})$ $= \{\mathsf{Source}(E_i) \mid E_i \in \{E_1, \ldots, E_n\}\} \cup \{\mathsf{Target}(E_i) \mid E_i \in \{E_1, \ldots, E_n\}\}$). Let $\mathsf{Cycles}(G)$ denote the set of cycles in G. Finally, if $E_1, \ldots, E_n$ is a cyclical path, then the* **length** *of the cycle is n.*

Example 11. *In Example 10, there are 3 cycles of length 1 (i.e. each of the nodes has a self-attack), 3 cycles of length 2 (i.e. each pair of nodes has a cycle), and 1 cycle of length 3 (i.e. there is a cycle involving all three nodes). So $\mathsf{Cycles}(G) = 7$.*

9

Definition 14. *A graph G is* **disjoint** *with graph G' iff*

$$\mathsf{Nodes}(G) \cap \mathsf{Nodes}(G') = \emptyset$$

Definition 15. *A graph G has an* **inverse graph** *defined as*

$$\mathsf{Invert}(G) = (\mathsf{Nodes}(G), \mathsf{InverseArcs}(G))$$

where $\mathsf{InverseArcs}(G) = \{(B, A) \mid (A, B) \in \mathsf{Arcs}(G)\}$.

Definition 16. *A graph G is* **isomorphic** *with graph G' iff there is a bijection $f : \mathsf{Nodes}(G) \to \mathsf{Nodes}(G')$ such that $(A, B) \in \mathsf{Arcs}(G)$ iff $(f(A), f(B)) \in \mathsf{Arcs}(G)$.*

The following definition identifies a graph as being connected when for each pair of nodes in the graph, there is a sequence of arcs connecting them.

Definition 17. *$G' \sqsubseteq G$ is* **connected** *iff for all $A_i, A_j \in \mathsf{Nodes}(G')$ there is a path from A_i to A_j in the graph $(\mathsf{Nodes}(G'), \mathsf{UndirectedArcs}(G'))$ where $\mathsf{UndirectedArcs}(G') = \{(A, B), (B, A) \mid (A, B) \in \mathsf{Arcs}(G')\}$.*

The following definition for a multi-node component is a variant of the usual definition for a component. It is a component that has at least two elements.

Definition 18. *G' is a* **multi-node component** *of G iff the following three conditions hold.*

1. *$G' \sqsubseteq G$ s.t. G' is connected*

2. *there is no G'' s.t. $G' \sqsubseteq G''$ and $G'' \sqsubseteq G$ and G'' is connected*

3. *$|\mathsf{Nodes}(G)| \geq 2$*

Let $\mathsf{Components}(G) = \{G' \sqsubseteq G \mid G'$ is a multi-node component of $G\}$.

Example 12. *In the following graph, there are three multi-node components (left, middle, right).*
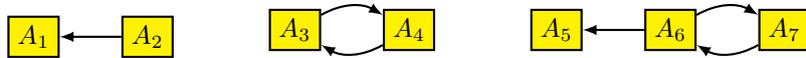
We introduce the definition for multi-node components because it forms the basis of a potentially useful measure of inconsistency that we will introduce in the next section. As we will see, we will not be interested in components that contain just 1 node.

3 Inconsistency measures for argument graphs

Following developments in inconsistency measures for logical knowledgebases, we define a graph-based inconsistency measure as a function that assigns a real number to each graph such that the following constraints of consistency and freeness are satisfied. We explain these constraints as follows: (Consistency) If a graph has no arcs, then the graph contains no counterarguments, and hence it is consistent; and (Freeness) Adding an argument that does not attack any argument or is not attacked by an argument does not change the inconsistency of the graph.

Definition 19. *A* **graph-based inconsistency measure** *is a function* $I : \mathcal{G} \to \mathbb{R}$ *such that*

1. *(Consistency) If* $\mathsf{Arcs}(G) = \emptyset$, *then* $I(G) = 0$.

2. *(Freeness) If* $\mathsf{Nodes}(G) = \mathsf{Nodes}(G') \setminus \{A\}$ *and* $\mathsf{Arcs}(G) = \mathsf{Arcs}(G')$, *then* $I(G) = I(G')$.

The following are further optional properties of a graph-based inconsistency measure: (Monotonicity) Adding arguments and counterarguments cannot decrease inconsistency; (Inversion) If G and G' have the same arguments, but each attack in G is reversed in G', then they have the same inconsistency measure; (Isomorphism) If G and G' have the same structure, then they have the same inconsistency measure; (Disjoint additivity) If G_1 and G_2 are disjoint, then the inconsistency of $G_1 + G_2$ is the sum of the inconsistency of G_1 and G_2; and (Super-additivity) The inconsistency measure of the union of G_1 and G_2 is not less than the sum of the inconsistency measure of G_1 and the inconsistency measure of G_2.

Definition 20. *The following are further properties for a graph-based inconsistency measure.*

- *(Monotonicity) If* $G \sqsubseteq G'$, *then* $I(G) \leq I(G')$.

- *(Inversion) If* $G' = \mathsf{Invert}(G)$, *then* $I(G) = I(G')$.

- *(Isomorphic invariance) If* G *and* G' *are isomorphic, then* $I(G) = I(G')$.

- *(Disjoint additivity) If* G_1 *and* G_2 *are disjoint, then* $I(G_1 + G_2) = I(G_1) + I(G_2)$.

- *(Super-additivity)* $I(G_1 + G_2) \geq I(G_1) + I(G_2)$.

In the following subsections, we consider two classes of inconsistency measure for abstract argumentation, namely graph structure measures, and graph extension measures. We will compare and contrast them using the above properties.

3.1 Graph structure measures

We now provide some proposals for measures. We give examples of them, and then show that they are graph-based inconsistency measures.

Definition 21. *The following are measures $I : \mathcal{G} \to \mathbb{R}$.*

- *(Drastic)*

$$\text{If } \mathsf{Arcs}(G) \neq \emptyset, \text{ then } I_{dr}(G) = 1, \text{ otherwise} I_{dr}(G) = 0$$

- *(InSum)*

$$I_{in}(G) = \sum_{A \in \mathsf{Nodes}(G)} \mathsf{Indegree}(G, A)$$

- *(WeightedInSum)*

$$I_{win}(G) = \sum_{A \in \mathsf{Nodes}(G) \ s.t. \ \mathsf{Indegree}(G,A) \geq 1} \frac{1}{\mathsf{Indegree}(G, A)}$$

- *(WeightedOutSum)*

$$I_{wou}(G) = \sum_{A \in \mathsf{Nodes}(G) \ s.t. \ \mathsf{Outdegree}(G,A) \geq 1} \frac{1}{\mathsf{Outdegree}(G, A)}$$

- *(CycleCount)*

$$I_{cc}(G) = |\mathsf{Cycles}(G)|$$

- *(WeightedCycleCount)*

$$I_{wcc}(G) = \sum_{C \in \mathsf{Cycles}(G)} \frac{1}{|C|}$$

- *(WeightedComponentCount)*

$$\text{If } \mathsf{Components}(G) = \emptyset, \text{ then } I_{ic}(G) = 0, \text{ otherwise}$$

$$I_{ic}(G) = \sum_{X \in \mathsf{Components}(G)} (|X| - 1)^2,$$

We explain these measures as follows: (Drastic) If a graph has attacks (i.e. counterarguments), then the inconsistency measure is 1, otherwise the inconsistency measure is 0; (InSum) This is the sum of the indegrees for the nodes in the graph; (WeightedInSum) This is the sum of the inverse of indegrees

for the nodes in the graph and so a node with a lower indegree has higher contribution to the inconsistency; (WeightedOutSum) This is the sum of the inverse of outdegrees for the nodes in the graph and so a node with a lower outdegree has a higher contribution to the inconsistency; (CycleCount) The is the number of the cycles in the graph; (WeightedCycleSum) This is the sum of the inverse of the number cycles in the graph and so a shorter cycle has a higher contribution to the inconsistency; and (WeightedComponentCount) This is the sum of the cardinality minus 1 squared of each component in the graph and so a larger component has a higher contribution to the inconsistency.
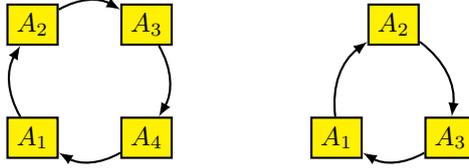
Example 13. *Consider the following graphs G_1 (left) and G_2 (right).*

Hence, we get the following inconsistency evaluations.

	I_{in}	I_{win}	I_{wou}	I_{cc}	I_{wcc}	I_{ic}
G_1	3	1/3	3	0	0	9
G_2	2	1/2	2	0	0	4

Example 14. *Consider the following graphs G_1 (left) and G_2 (right).*

Hence, we get the following inconsistency evaluations.

	I_{in}	I_{win}	I_{wou}	I_{cc}	I_{wcc}	I_{ic}
G_1	4	4	4	1	1/4	9
G_2	3	3	3	1	1/3	4

The graph structure measures are graph-based inconsistency measures as shown in the following result.

Proposition 1. *The I_{dr}, I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc} and I_{ic} measures are graph-based inconsistency measures according to Definition 19.*

Proof. Follows directly from definitions. $\square$

The following result gives some idea of the difference of scale for each measure.

Proposition 2. *If G is a complete graph and $|\mathsf{Nodes}(G)| = n$ where $n > 0$, then*

$$
\begin{array}{ll}
I_{dr} = 1 & I_{in} = n^2 \\
I_{win} = 1 & I_{wou} = 1 \\
I_{cc} = 2^n - 1 & I_{wcc} = \sum_{k=1}^{n} \left[\frac{1}{k} \times \frac{n!}{k!(n-k)!} \right] \\
I_{ic} = (n-1)^2 &
\end{array}
$$

Proof. Assume G is a complete graph and $|\mathsf{Nodes}(G)| = n$. (I_{dr}) $\mathsf{Arcs}(G) \neq \emptyset$, and so $I_{dr} = 1$. (I_{in}) For every node A, $\mathsf{Indegree}(G, A) = n$, and so $I_{in} = n^2$. (I_{win}) For every node A, $\mathsf{Indegree}(G, A) = n$, and so $I_{win}(G) = \sum_{A \in \mathsf{Nodes}(G)} \frac{1}{\mathsf{Indegree}(G,A)} = \sum_{A \in \mathsf{Nodes}(G)} \frac{1}{n} = n \times \frac{1}{n} = 1$. (I_{wou}) Ditto. (I_{cc}) There are $2^n - 1$ subsets of nodes where each subset constitutes a cycle, and so $I_{cc} = 2^n - 1$. (I_{wcc}) For each non-empty subset of nodes constitute a cycle. Furthermore, for each $k \in \{1, \ldots, n\}$, there are $\frac{n!}{k!(n-k)!}$ subsets of cardinality k, and each of these contribute $\frac{1}{k}$ to the sum. Hence, the sum is $\sum_{k=1}^{n} \left[\frac{1}{k} \times \frac{n!}{k!(n-k)!} \right]$. (I_{ic}) There is a single component, and so $I_{ic} = (n-1)^2$. $\qquad \square$

The following result shows which optional properties are satisfied by which measures.

Proposition 3. *For the graph structure measures in Definition 21, the adherence to the properties in Definition 20 is summarized in the following table.*

	I_{dr}	I_{in}	I_{win}	I_{wou}	I_{cc}	I_{wcc}	I_{ic}
Monotonicity	✓	✓	✓	✓	✓	✓	✓
Inversion	✓	✓	×	×	✓	✓	✓
Isomorphic invariance	✓	✓	✓	✓	✓	✓	✓
Disjoint additivity	×	✓	✓	✓	✓	✓	✓
Super-additivity	×	✓	×	×	✓	✓	×

Proof. We consider each property as follows.

- Monotonicity. $(I_{dr},\ I_{in},\ I_{win},\ I_{wou},\ I_{cc},\ I_{wcc}, I_{ic})$ Follows directly from the definition.

- Inversion. $(I_{dr},\ I_{in})$ Follows directly from the definition. $(I_{win},\ I_{wou})$ Consider G (left) and $\mathsf{Invert}(G)$ (right). Therefore, we obtain $I_{win}(G) = 1/2$, $I_{win}(\mathsf{Invert}(G)) = 2$, $I_{wou}(G) = 2$, and $I_{wou}(\mathsf{Invert}(G)) = 1/2$.

$(I_{cc},\ I_{wcc},\ I_{ic})$ Follows directly from the definition.

- Isomorphic invariance. $(I_{dr}, I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc}, I_{ic})$ Follows directly from definition.

- Disjoint additivity. (I_{dr}) Consider the following graphs G_1 (left) and G_2 (right) where $I_{dr}(G_1) = 1$ and $I_{dr}(G_2) = 1$ but $I_{dr}(G_1 + G_2) = 1$.

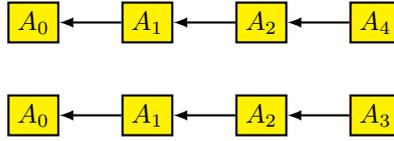$$A_1 \longleftarrow A_2 \qquad A_3 \longleftarrow A_4$$

$(I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc}, I_{ic})$ Follows directly from the definition.

- Super-additivity. (I_{dr}) See counterexample for disjoint additivity. (I_{in}) Follows directly from the definition. (I_{win}) Consider the following graphs G_1 (left) and G_2 (right) where $I_{win}(G_1) = 1$ and $I_{win}(G_2) = 1$ but $I_{win}(G_1 + G_2) = 1/2$.

$$A_1 \longleftarrow A_2 \qquad A_1 \longleftarrow A_3$$

(I_{wou}) Use a similar counterexample to I_{win}. (I_{cc}, I_{wcc}) Follows directly from the definition. (I_{ic}) Consider the following graphs G_1 (top) and G_2 (bottom) where $I_{ic}(G_1) = 9$ and $I_{ic}(G_2) = 9$ but $I_{ic}(G_1 + G_2) = 16$.

$$A_0 \longleftarrow A_1 \longleftarrow A_2 \longleftarrow A_4$$

$$A_0 \longleftarrow A_1 \longleftarrow A_2 \longleftarrow A_3$$

$\square$

The following property of order-compatibility holds when two measures give the same ranking to all the graphs. If this holds, then there is some overlap in what the two measures offer.

Definition 22. *Measures I_x and I_y are* **order-compatible** *if for all G_1 and G_2,*
$$I_x(G_1) < I_x(G_2) \text{ iff } I_y(G_1) < I_y(G_2)$$
otherwise I_x and I_y are **order-incompatible**.

Proposition 4. *The $I_{dr}, I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc}$ and I_{ic} measures are pairwise order-incompatible,*

Proof. From the differences in satisfaction of properties in Proposition 3, I_{dr} is pairwise incompatible with the other measures, I_{ic} is pairwise incompatible with the other measures, and I_{win} and I_{wou} are pairwise incompatible with the other measures, though from the properties in Proposition 3, we cannot discriminate I_{win} from I_{wou}. However, we can discriminate I_{win} from I_{wou} with graph G_1 (left) and G_2 (right), where $I_{win}(G_1) = 1/2$, $I_{win}(G_2) = 2$, $I_{wou}(G_1) = 2$, and $I_{wou}(G_2) = 1/2$.

Finally, from the properties in Proposition 3, we cannot discriminate between I_{in}, I_{wcc}, I_{cc}. So, from Proposition 2, assuming G is complete, if $|\mathsf{Nodes}(G)| \in \{2, 3, 4\}$, then $I_{in}(G) > I_{cc}(G)$, whereas if $|\mathsf{Nodes}(G)| \geq 5$, then $I_{in}(G) < I_{cc}(G)$. Similarly, if $|\mathsf{Nodes}(G)| = 2$, then $I_{in}(G) > I_{wcc}(G)$, whereas if $|\mathsf{Nodes}(G)| = 10$, then $I_{in}(G) < I_{wcc}(G)$. For I_{cc} and I_{wcc}, consider G_1 (left) and G_2 (right) which give $I_{cc}(G_1) = I_{cc}(G_2)$ and $I_{wcc}(G_1) > I_{wcc}(G_2)$.

$\square$

In this section, we have considered a range of measures that take the structure of the argument graph into account. Each has its rationale, and any combination of them may provide useful insights into the nature of the conflict in an argumentation scenario.

3.2 Graph extension measures

We now consider measures of inconsistency that take the extensions of the graph into account. In the following, we consider three measures: (1) I_{pr} which gives the number of preferred extensions minus 1; (2) I_{ngr} which gives the number of arguments not in the grounded extension and not attacked by a member of the grounded extension; and (3) I_{nst} which gives the minimum number of arguments to be removed to get a stable extension.

Definition 23. *The following are measures $I : \mathcal{G} \to \mathbb{R}$.*

- *(PreferredCount)*

$$I_{pr}(G) = |\mathsf{Extensions}_{\mathsf{pr}}(G)| - 1$$

- *(NonGroundedCount)*

$$I_{ngr}(G) = |\mathsf{Nodes}(G) \setminus (\mathsf{Extensions}_{\mathsf{gr}}(G) \cup \mathsf{Attackees}(G))|$$

where $\mathsf{Attackees}(G) = \{B \mid (A, B) \in \mathsf{Arcs}(G) \ \text{and} \ A \in \mathsf{Extensions}_{\mathsf{gr}}(G)\}$.
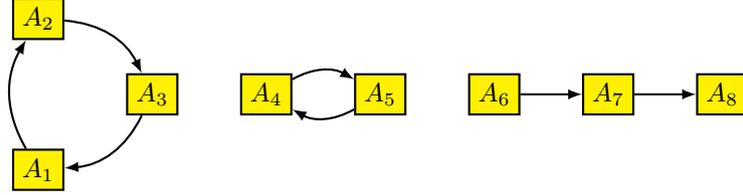
- *(UnstableCount)*

$$I_{ust}(G) = \min\{|X| \mid \mathsf{Extensions_{st}}(\mathsf{Induce}(G, X)) \neq \emptyset \ s.t. \ X \subseteq \mathsf{Nodes}(G)\}$$

Example 15. *For the following argument graph, we have the following extensions,*

- $\mathsf{Extensions_{pr}}(G) = \{\{A_4, A_6, A_8\}, \{A_5, A_6, A_8\}\}$

- $\mathsf{Extensions_{gr}}(G) = \{\{A_6, A_8\}\}$

and removing one argument from $\{A_1, A_2, A_3\}$ is the smallest number of arguments to be removed to get a stable extension. Hence, $I_{pr}(G) = 1$, $I_{ngr}(G) = 5$, and $I_{ust}(G) = 1$.

By way of further explanation for the I_{ust} measure, a stable extension is a preferred extension where every argument not in the extension is attacked by an argument that is in the extension. Not every graph has a stable extension. But if arguments and the attacks involving those arguments are removed, then a stable extension can be obtained from this subgraph. So this measure is based on the minimum number of arguments that need to be removed to get a stable extension.

The following result shows that the extension-based measures are indeed inconsistency measures but the subsequent result shows that most of the optional properties do not hold for these measures.

Proposition 5. *The I_{pr}, I_{ngr}, and I_{ust} measures are graph-based inconsistency measures according to Definition 19.*

Proof. We show the satisfaction of consistency by assuming that $\mathsf{Arcs}(G) = \emptyset$: (I_{pr}) $\mathsf{Extensions_{pr}}(G) = \{\mathsf{Nodes}(G)\}$, and so $I_{pr}(G) = 0$; $(I_{ngr})\mathsf{Extensions_{gr}}(G) = \{\mathsf{Nodes}(G)\}$, and so $I_{ngr}(G) = 0$; (I_{ust}) $\mathsf{Extensions_{st}}(G) \neq \emptyset$, and so the smallest X such that $\mathsf{Extensions_{st}}(\mathsf{Induce}(G, X)) \neq \emptyset$ is $X = \emptyset$, and so $I_{ust}(G) = 0$. We show the satisfaction of freeness by assuming that $\mathsf{Nodes}(G) = \mathsf{Nodes}(G') \setminus \{A\}$ and $\mathsf{Arcs}(G) = \mathsf{Arcs}(G')$: (I_{pr}) $|\mathsf{Extensions_{gr}}(G)| = |\mathsf{Extensions_{gr}}(G')|$, and so $I_{pr}(G) = I_{pr}(G')$; (I_{ngr}) Let X be $\mathsf{Nodes}(G') \setminus \mathsf{Nodes}(G)$, $\mathsf{Extensions_{pr}}(G) = \{E\}$, and $\mathsf{Extensions_{pr}}(G') = \{E'\}$, and so from the assumptions, $E = E' \setminus X$, and hence $\mathsf{Nodes}(G) \setminus (E \cup \mathsf{Attackees}(G)) = \mathsf{Nodes}(G') \setminus (E' \cup \mathsf{Attackees}(G'))$,

and so $I_{ngr}(G) = I_{ngr}(G')$; (I_{ust}) Let X_1 be the smallest subset of nodes such that $\mathsf{Extensions_{st}}(\mathsf{Induce}(G, X_1)) \neq \emptyset$. From the assumptions, X_1 is also the smallest subset of nodes such that $\mathsf{Extensions_{st}}(\mathsf{Induce}(G', X_1)) \neq \emptyset$. Hence, $I_{ust}(G) = I_{ust}(G')$. $\qquad\square$

Proposition 6. *For the graph structure measures in Definition 21, the adherence to the properties in 20 is summarized in the following table.*

	I_{pr}	I_{ngr}	I_{ust}
Monotonicity	×	×	×
Inversion	×	×	×
Isomorphic invariance	✓	✓	✓
Disjoint additivity	×	✓	✓
Super-additivity	×	×	×

Proof. We consider each property as follows.

- Monotonicity. (I_{pr}) Consider G_1 (left) and G_2 (right) where $I_{pr}(G_1) = 1$, $I_{pr}(G_2) = 0$, and $G_1 \sqsubseteq G_2$.

  (I_{ngr}) Consider G_1 (above left) and G_2 (above right) where $I_{ngr}(G_1) = 2$, $I_{ngr}(G_2) = 0$, and $G_1 \sqsubseteq G_2$. (I_{ust}) Consider G_1 (left) and G_2 (right) where $I_{ust}(G_1) = 1$, $I_{ust}(G_2) = 0$, and $G_1 \sqsubseteq G_2$.

- Inversion. (I_{pr}) Consider G (left) and $\mathsf{Inverted}(G)$ with $I_{pr}(G) = 0$, and $I_{pr}(\mathsf{Inverted}(G)) = 1$.

  (I_{ngr}) Consider G (above left) and $\mathsf{Inverted}(G)$ (above right) with $I_{ngr}(G) = 0$, and $I_{ngr}(\mathsf{Inverted}(G)) = 3$. ($I_{ust}$) Consider G (left) and $\mathsf{Inverted}(G)$ (right) with $I_{ust}(G) = 0$, and $I_{ust}(\mathsf{Inverted}(G)) = 1$.

- Isomorphic invariance. (I_{pr},I_{ngr},I_{ust}) Follows directly from definition.

18

- Disjoint additivity. (I_{pr}) Consider G_1 (left) and G_2 (right) where $I_{pr}(G_1)$ = 1, $I_{pr}(G_2) = 1$, and $I_{pr}(G_1 + G_2) = 3$.

(I_{ngr}) If G_1 & G_2 are disjoint, then $\mathsf{Extensions_{gr}}(G_1) \cap \mathsf{Extensions_{gr}}(G_2) = \emptyset$, $\mathsf{Attackees_{gr}}(G_1) \cap \mathsf{Attackees_{gr}}(G_2) = \emptyset$, and $\mathsf{Nodes_{gr}}(G_1) \cap \mathsf{Nodes_{gr}}(G_2) = \emptyset$. So $I_{ngr}(G_1 + G_2) = I_{ngr}(G_1) + I_{ngr}(G_2)$. (I_{ust}) If G_1 and G_2 are disjoint, $\min\{|X| \mid \mathsf{Extensions_{st}}(\mathsf{Induce}(G_1 + G_2, X)) \neq \emptyset$ s.t. $X \subseteq \mathsf{Nodes}(G_1 + G_2)\} = \min\{|X| \mid \mathsf{Extensions_{st}}(\mathsf{Induce}(G_1, X)) \neq \emptyset$ s.t. $X \subseteq \mathsf{Nodes}(G_1)\} + \min\{|X| \mid \mathsf{Extensions_{st}}(\mathsf{Induce}(G_2, X)) \neq \emptyset$ such that $X \subseteq \mathsf{Nodes}(G_2)\}$. Therefore, $I_{ust}(G_1 + G_2) = I_{ust}(G_1) + I_{ust}(G_2)$.

- Super-additivity. (I_{pr}) Consider G_1 (left) and G_2 (right) where $I_{pr}(G_1) = 1$, $I_{pr}(G_2) = 0$, and $I_{pr}(G_1 + G_2) = 0$.

(I_{ngr}) Consider G_1 (above left) and G_2 (above right) where $I_{ngr}(G_1) = 2$, $I_{ngr}(G_2) = 0$, and $I_{ngr}(G_1 + G_2) = 0$. (I_{ust}) Consider G_1 (left) and G_2 (right) where $I_{ust}(G_1) = 1$, $I_{ust}(G_2) = 0$, and $I_{ust}(G_1 + G_2) = 0$.

$\square$

Proposition 7. *The I_{pr}, I_{ngr}, I_{ust} measures are pairwise order-incompatible. Each is also pairwise order-incompatible with each of the I_{dr}, I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc} and I_{ic} measures.*

Proof. From the differences in satisfaction of properties in Proposition 3, and Proposition 6, the extension-based measures are pairwise incompatible with the structure-based measures. We now consider discriminating between the extension-based measures. We can discriminate I_{pr} from I_{ngr} with graph G_1 (left) and G_2 (right), where $I_{pr}(G_1) = 1$, $I_{pr}(G_2) = 0$, $I_{ngr}(G_1) = 2$, and $I_{ngr}(G_2) = 3$.

19

We can discriminate I_{pr} from I_{ust} with graph G_1 (left) and G_2 (right), where $I_{pr}(G_1) = 1$, $I_{pr}(G_2) = 3$, $I_{ust}(G_1) = 2$, and $I_{ust}(G_2) = 0$.

We can discriminate I_{pr} from I_{ust} with G_1 (left) and G_2 (right), where $I_{ngr}(G_1)$ $= 1$, $I_{ngr}(G_2) = 2$, $I_{ust}(G_1) = 1$, and $I_{ust}(G_2) = 1$.

$\square$

In this subsection, we have considered an alternative class of measures based on the extensions of the graph. As can be seen from the properties that hold for them, they are very different in behaviour from the structured-based measures. On the one hand, measuring inconsistency in terms of the extensions seems a natural option, but on the other hand, they fail to satisfy most of the optional properties that we have considered, though this may be more a reflection of the nature of these properties, and that we need to consider a wider range of properties to justifiy and characterise extension-based measures.

4 Logic-based instantiations

We now consider structured argumentation. This is where the arguments in the argument graph are instantiated with logical structure. We proceed by reviewing a specific approach to structured argumentation called deductive argumentation, and then we consider two approaches to measuring inconsistency in deductive argumentation. The first is the degree of undercut approach proposed in [BH05, BH08] and the second is the application of existing logic-based measures of inconsistency to deductive arguments in an argument graph.

4.1 Review of deductive argumentation

In deductive argumentation, each argument is a pair where the first item is a set of premises that logically entails the second item according to some logical consequence relation (for a review see [BH14]). So we have a logical language to express the set of premises and the claim, and we have a logical consequence relation to relate the premises to the claim. So in order to construct argument graphs with deductive arguments, we need to specify the choice of logic (which we call the base logic) that we use to define arguments and counterarguments.

Classical logic is appealing as the choice of base logic as it reflects the richer deductive reasoning often seen in arguments arising in discussions and debates. We assume the usual propositional and predicate (first-order) languages for classical logic, and the usual **classical consequence relation**, denoted $\vdash$. A **knowledgebase** is a set of classical propositional or predicate formulae.

Definition 24. *For a classical knowledgebase Φ, and a classical formula α, $\langle\Phi, \alpha\rangle$ is a* **classical argument** *iff $\Phi \vdash \alpha$ and $\Phi \nvdash \bot$ and there is no proper subset Φ' of Φ such that $\Phi' \vdash \alpha$. For an argument $A = \langle\Phi, \alpha\rangle$, the function* $\mathsf{Support}(A)$ *returns Φ and the function* $\mathsf{Claim}(A)$ *returns α.*

Example 16. *The following classical argument uses a universally quantified formula in contrapositive reasoning to obtain the claim about number 77.*

$$\langle\{\forall\mathtt{X.multipleOfTen(X)} \to \mathtt{even(X)}, \neg\mathtt{even(77)}\}, \neg\mathtt{multipleOfTen(77)}\rangle$$

A counterargument is an argument that attacks another argument. In deductive argumentation, we define the notion of counterargument in terms of logical contradiction between the claim of the counterargument and the premises or claim of the attacked argument. Given the expressivity of classical logic (in terms of language and inferences), there are a number of natural ways to define counterarguments (as given in Definition 25 below). For instance, the definition of rebuttal captures the situation where two arguments have complementary claims, whereas the definition of defeater captures the situation where the claim of one argument negates the support of the other argument.

Definition 25. *Let A and B be two classical arguments. We define the following types of* **attack**. *In each case, A is the* **attacker** *and B is the* **attackee**.

- *A is a* **defeater** *of B if* $\mathsf{Claim}(A) \vdash \neg \bigwedge \mathsf{Support}(B)$.

- *A is a* **direct defeater** *of B if $\exists\phi \in \mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A) \vdash \neg\phi$.*

- *A is an* **undercut** *of B if $\exists\Psi \subseteq \mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A) \equiv \neg\bigwedge\Psi$.*

- *A is a* **direct undercut** *of B if $\exists\phi \in \mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A) \equiv \neg\phi$.*

- *A is a* **canonical undercut** *of B if $\mathsf{Claim}(A) \equiv \neg\bigwedge\mathsf{Support}(B)$.*

- *A is a* **rebuttal** *of B if $\mathsf{Claim}(A) \equiv \neg\mathsf{Claim}(B)$.*

- *A is a* **defeating rebuttal** *of B if $\mathsf{Claim}(A) \vdash \neg\mathsf{Claim}(B)$.*

To illustrate these different notions of counterargument, we consider the following examples, and we relate these definitions in Figure 1 where we show that defeaters are the most general of these definitions.
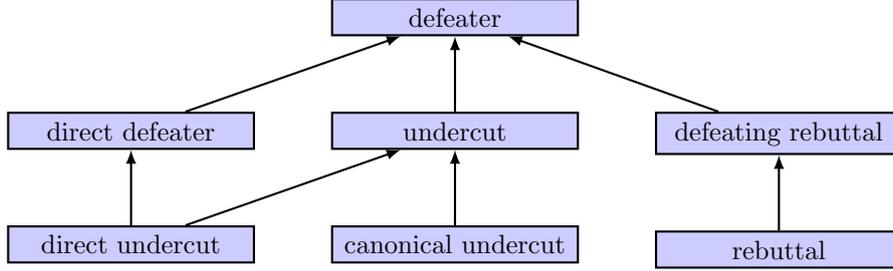
Figure 1: We can represent the containment between the attack relations as above where an arrow from R_1 to R_2 indicates that $R_1 \subseteq R_2$.

Example 17. *Let $\{a \vee b, a \leftrightarrow b, c \to a, \neg a \wedge \neg b, a, b, c, a \to b, \neg a, \neg b, \neg c\}$ be the knowledgebase from which the following are examples of attacks.*

$\langle \{a \vee b, c\}, (a \vee b) \wedge c \rangle$ *is a defeater of* $\langle \{\neg a, \neg b\}, \neg a \wedge \neg b \rangle$
$\langle \{a \vee b, c\}, (a \vee b) \wedge c \rangle$ *is a direct defeater of* $\langle \{\neg a \wedge \neg b\}, \neg a \wedge \neg b \rangle$
$\langle \{\neg a \wedge \neg b\}, \neg(a \wedge b) \rangle$ *is a undercut of* $\langle \{a, b, c\}, a \wedge b \wedge c \rangle$
$\langle \{\neg a \wedge \neg b\}, \neg a \rangle$ *is a direct undercut of* $\langle \{a, b, c\}, a \wedge b \wedge c \rangle$
$\langle \{\neg a \wedge \neg b\}, \neg(a \wedge b \wedge c) \rangle$ *is a canonical undercut of* $\langle \{a, b, c\}, a \wedge b \wedge c \rangle$
$\langle \{a, a \to b\}, b \vee c \rangle$ *is a rebuttal of* $\langle \{\neg a \wedge \neg b, \neg c\}, \neg(b \vee c) \rangle$
$\langle \{a, a \to b\}, b \rangle$ *is a defeating rebuttal of* $\langle \{\neg a \wedge \neg b, \neg c\}, \neg(b \vee c) \rangle$

Example 18. *Consider the following arguments where A_2 is an undercut of A_1 though it is neither a direct undercut nor a canonical undercut. Essentially, the attack says that the flight cannot be both a low cost flight and a luxury flight.*

$A_1 = \langle \{\texttt{lowCostFly}, \texttt{luxFly}, \texttt{lowCostFly} \wedge \texttt{luxFly} \to \texttt{goodFly}\}, \texttt{goodFly} \rangle$
$A_2 = \langle \{\neg\texttt{lowCostFly} \vee \neg\texttt{luxFly}\}, \neg\texttt{lowCostFly} \vee \neg\texttt{luxFly} \rangle$

Trivially, undercuts are defeaters but it is also quite simple to establish that rebuttals are defeaters. Furthermore, if an argument has defeaters then it has undercuts. It may happen that an argument has defeaters but no rebuttals as illustrated next.

Example 19. *Consider* $\{\neg\texttt{containsGarlic} \wedge \texttt{goodDish}, \neg\texttt{goodDish}\}$ *as the knowledgebase. Then the following argument has at least one defeater but no rebuttal.*

$$\langle \{\neg\texttt{containsGarlic} \wedge \texttt{goodDish}\}, \neg\texttt{containsGarlic} \rangle$$

There are some important differences between rebuttals and undercuts that can be seen in the following examples.
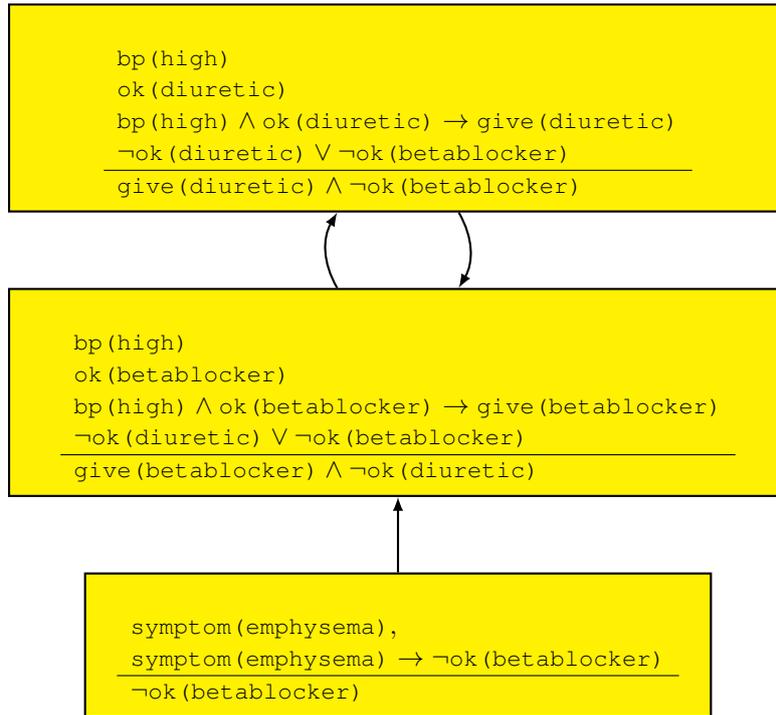
22

Figure 2: An instantiated argument graph for the abstract argument graph in Example 1. The atom bp(high) denotes that the patient has high blood pressure. The top two arguments rebut each other (i.e. the attack is defeating rebut). For this, each argument has an integrity constraint in the premises that says that it is not ok to give both betablocker and diuretic. So the top argument is attacked on the premise ok(diuretic) and the middle argument is attacked on the premise ok(betablocker). So we are using the ok predicate as a normality condition for the rule to be applied.

Example 20. *Consider the following arguments. The first argument A_1 is a direct undercut to the second argument A_2, but neither rebuts each other. Furthermore, A_1 "agrees" with the claim of A_2 since the premises of A_1 could be used for an alternative argument with the same claim as A_2.*

$A_1 = \langle \{\neg\texttt{containsGarlic} \wedge \neg\texttt{goodDish}\}, \neg\texttt{containsGarlic}\rangle$
$A_2 = \langle \{\texttt{containsGarlic}, \texttt{containsGarlic} \rightarrow \neg\texttt{goodDish}\}, \neg\texttt{goodDish}\rangle$

Example 21. *Consider the following arguments. The first argument is a rebuttal of the second argument, but it is not an undercut because the claim of the first argument is not equivalent to the negation of some subset of the premises of the second argument.*

$A_1 = \langle \{\texttt{goodDish}\}, \texttt{goodDish}\rangle$
$A_2 = \langle \{\texttt{containsGarlic}, \texttt{containsGarlic} \rightarrow \neg\texttt{goodDish}\}, \neg\texttt{goodDish}\rangle$
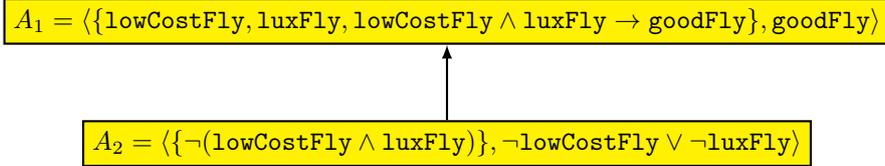
So an undercut for an argument need not be a rebuttal for that argument, and a rebuttal for an argument need not be an undercut for that argument.

An instantiated argument graph is an argument graph where each node is a classical argument, and each arc is an attack conforming to the definitions for attack (Definition 25). We provide illustrations of instantiated argument graphs in the following example and in Figure 2.

Example 22. *Consider the following argument graph where A_1 is "The flight is low cost and luxury, therefore it is a good flight", and A_2 is "A flight cannot be both low cost and luxury".*

$$A_1 \longleftarrow A_2$$

For this, we instantiate the arguments in the above abstract argument graph with arguments we considered in Example 18. This gives us the following instantiated argument graph where A_2 is an undercut to A_1.

$$A_1 = \langle \{\texttt{lowCostFly}, \texttt{luxFly}, \texttt{lowCostFly} \wedge \texttt{luxFly} \rightarrow \texttt{goodFly}\}, \texttt{goodFly}\rangle$$

$$\uparrow$$

$$A_2 = \langle \{\neg(\texttt{lowCostFly} \wedge \texttt{luxFly})\}, \neg\texttt{lowCostFly} \vee \neg\texttt{luxFly}\rangle$$

Perhaps the first paper to consider instantiating argument graphs with deductive arguments based on classical logic is by Cayrol [Cay95] using direct undercut. For more details on deductive argumentation and how it can be used to instantiate argument graphs, see [BH14].

4.2 Degree of undercut

An argument conflicts with each of its undercuts, by the very definition of an undercut. Now, some may conflict more than others, and some may conflict a little while others conflict a lot.

Example 23. *Consider the following argument graph G. Each undercut has a premise that negates some or all of the premises in the root. The left child has the weakest premise which can be read as saying that one of the premises in the root is false without saying which, the middle child says that one of the premises in the root is false and states which one, and the right child says that all of the premises are false.*
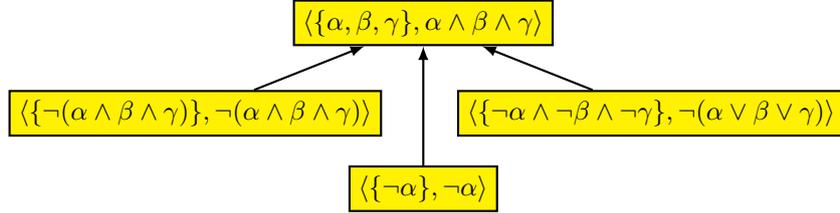
By these simple examples of undercuts, we see that there can be a difference in the amount of conflict between supports, and hence can be taken as an intuitive starting point for defining the degree of undercut that an argument has against its parent. To address this, the degree of undercut is a measure of the conflict between a pair of arguments based on the supports of these arguments [BH05, BH08]. There are some alternatives for defining the degree of undercut, and we review one of these proposals in the next subsection.

4.2.1 Degree of undercut based on Dalal distance

In this section, we investigate a degree of undercut based on the distance between pairs of models. For this, we use the Dalal distance [Dal88].

Definition 26. *For the language $\mathcal{L}$, let $\mathsf{Atoms}(\mathcal{L})$ be the set of atoms used in the language (and so the formulae in $\mathcal{L}$ are composed from $\mathsf{Atoms}(\mathcal{L})$ and the logical connectives using the usual inductive definition).*

Definition 27. *Let Π be a finite non-empty subset of $\mathsf{Atoms}(\mathcal{L})$ and let $w_i, w_j \in \wp(\Pi)$. The **Dalal distance** between w_i and w_j, denoted $\mathsf{Dalal}(w_i, w_j)$, is the difference in the number of atoms assigned true:*

$$\mathsf{Dalal}(w_i, w_j) = |w_i - w_j| + |w_j - w_i|$$

Example 24. *Let $w_1 = \{\alpha, \gamma, \delta\}$ and $w_2 = \{\beta, \gamma\}$ where $\{\alpha, \beta, \gamma, \delta\} \subseteq \Pi$. Then,*

$$\mathsf{Dalal}(w_1, w_2) = |\{\alpha, \delta\}| + |\{\beta\}| = 3$$

To evaluate the conflict between the support of an argument A and the support of an undercut A', we consider the models of $\mathsf{Support}(A)$ and $\mathsf{Support}(A')$, restricted to a set of atoms Π. For this, we require the following definition.

Definition 28. *Let Π is a finite non-empty subset of $\mathsf{Atoms}(\mathcal{L})$, let Φ be a set of formulae, and let $\models$ be the classical satisfaction relation.*

$$\mathsf{Models}(\Phi, \Pi) = \{w \in \wp(\Pi) \mid \forall \phi \in \Phi \text{ and } w \models \phi\}$$

Example 25. *Let $\Phi = \{\alpha \wedge \delta, \neg\phi, \gamma \vee \delta, \neg\psi, \beta \vee \gamma\} \subseteq \Delta$, and let Π be $\{\alpha, \beta, \gamma, \delta, \phi\} \subseteq \mathsf{Atoms}(\mathcal{L})$. Should it be the case that $\Pi \neq \mathsf{Atoms}(\mathcal{L})$, the set of all the models of Φ is a proper superset of $\mathsf{Models}(\Phi, \Pi)$ as the latter consists exactly of the following models.*

$$\{\alpha, \beta, \gamma, \delta\} \qquad \{\alpha, \beta, \delta\} \qquad \{\alpha, \gamma, \delta\}$$

To evaluate the conflict between two sets of formulae, we take a pair of models restricted to Π, one for each set, such that the Dalal distance is minimized. The degree of conflict is this distance divided by the maximum possible Dalal distance between a pair of models (i.e. $\log_2$ of the total number of models in $\wp(\Pi)$ which is $|\Pi|$).

Definition 29. *The **degree of conflict** wrt Π, denoted $\mathsf{Conflict}(\Phi, \Psi, \Pi)$, is:*

$$\frac{\min\{\mathsf{Dalal}(w_\Phi, w_\Psi) \mid w_\Phi \in \mathsf{Models}(\Phi, \Pi), w_\Psi \in \mathsf{Models}(\Psi, \Pi)\}}{|\Pi|}$$

Example 26. *Let $\Pi = \{\alpha, \beta, \gamma, \delta\}$.*

$$\mathsf{Conflict}(\{\alpha \wedge \beta \wedge \gamma \wedge \delta\}, \{\neg\alpha \vee \neg\beta \vee \neg\gamma\}, \Pi) = 1/4$$
$$\mathsf{Conflict}(\{\alpha \wedge \beta \wedge \gamma \wedge \delta\}, \{\neg(\alpha \vee \beta)\}, \Pi) = 2/4$$
$$\mathsf{Conflict}(\{\alpha \wedge \beta \wedge \gamma \wedge \delta\}, \{\neg\alpha \wedge \neg\beta \wedge \neg\gamma\}, \Pi) = 3/4$$

We obtain a degree of undercut by applying $\mathsf{Conflict}$ to supports as defined next.

Definition 30. *Let $A_i = \langle \Phi_i, \alpha_i \rangle$ and $A_j = \langle \Phi_j, \alpha_j \rangle$ be arguments.*
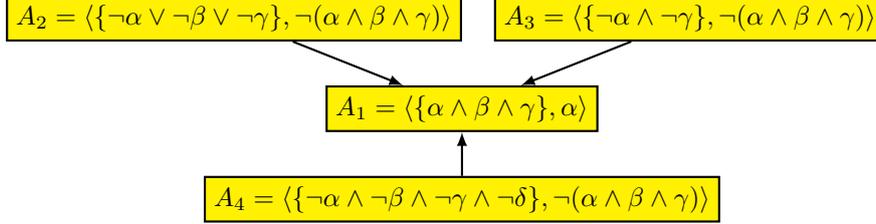
$$\mathsf{Degree}(A_i, A_j) = \mathsf{Conflict}(\Phi_i, \Phi_j, \mathsf{Atoms}(\Phi_i \cup \Phi_j))$$

Clearly, if A_i is an undercut for A_j, then $\mathsf{Degree}(A_i, A_j) > 0$.

Example 27. *Consider an argument with premises $\{\alpha, \beta, \gamma\}$. The claim is not important for the example. Values for degree of undercut by canonical undercuts are given below.*

$$\mathsf{Degree}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha \wedge \neg\beta \wedge \neg\gamma\}, \ldots\rangle) = 1$$
$$\mathsf{Degree}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha \wedge \neg\beta\}, \ldots\rangle) = 2/3$$
$$\mathsf{Degree}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha \vee \neg\beta \vee \neg\gamma\}, \ldots\rangle) = 1/3$$
$$\mathsf{Degree}(\langle\{\alpha, \beta, \gamma\}, \ldots\rangle, \langle\{\neg\alpha\}, \ldots\rangle) = 1/3$$

Example 28. *Consider the following argument graph where* $\mathsf{Degree}(A_1, A_2) = 1/3$, $\mathsf{Degree}(A_1, A_3) = 2/3$, *and* $\mathsf{Degree}(A_1, A_4) = 3/4$.

$$A_2 = \langle \{\neg\alpha \vee \neg\beta \vee \neg\gamma\}, \neg(\alpha \wedge \beta \wedge \gamma) \rangle \qquad A_3 = \langle \{\neg\alpha \wedge \neg\gamma\}, \neg(\alpha \wedge \beta \wedge \gamma) \rangle$$

$$A_1 = \langle \{\alpha \wedge \beta \wedge \gamma\}, \alpha \rangle$$

$$A_4 = \langle \{\neg\alpha \wedge \neg\beta \wedge \neg\gamma \wedge \neg\delta\}, \neg(\alpha \wedge \beta \wedge \gamma) \rangle$$

Example 29. *As a more general example, let* $A_1 = \langle \{\neg(\alpha_1 \vee \ldots \vee \alpha_n)\}, \neg(\alpha_1 \wedge \ldots \wedge \alpha_n) \rangle$, $A_2 = \langle \{\neg\alpha_1 \vee \ldots \vee \neg\alpha_n\}, \neg(\alpha_1 \wedge \ldots \wedge \alpha_n) \rangle$, $A_3 = \langle \{\neg\alpha_1\}, \neg(\alpha_1 \wedge \ldots \wedge \alpha_n) \rangle$, $A_4 = \langle \{\alpha_1 \wedge \ldots \wedge \alpha_n\}, \alpha_1 \rangle$.

$$\mathsf{Degree}(A_4, A_1) = n/n$$
$$\mathsf{Degree}(A_4, A_2) = 1/n$$
$$\mathsf{Degree}(A_4, A_3) = 1/n$$

The above examples indicate how the Degree measure differentiates between different kinds of attack, and the following result shows the Degree measure has the basic properties that we require.

Proposition 8. *Let* $A_i = \langle \Phi_i, \alpha_i \rangle$ *and* $A_j = \langle \Phi_j, \alpha_j \rangle$ *be arguments.*

$$\begin{aligned}
&(1) \quad 0 \leq \mathsf{Degree}(A_i, A_j) \leq 1 \\
&(2) \quad \mathsf{Degree}(A_i, A_j) = \mathsf{Degree}(A_j, A_i) \\
&(3) \quad \mathsf{Degree}(A_i, A_j) = 0 \ \textit{iff} \ \Phi_i \cup \Phi_j \not\vdash \bot
\end{aligned}$$

So the degree of undercut gives a value in the unit interval to represent how much two arguments differ in terms of their premises. In addition to this proposal, [BH08] presents some alternative proposals for defining degree of undercut.

4.2.2 Cumulative degree of undercut

We now consider how we can harness the notion of degree of undercut as an inconsistency measure for an argument graph.

Definition 31. *Let* G *be an argument graph. The* **cumulative degree of undercut** *in* G, *denoted* $\mathsf{I_{cu}}(G)$, *is given by*

$$I_{cu}(G) = \sum_{(A_i, A_j) \in \mathsf{Arcs}(G)} \mathsf{Degree}(A_i, A_j)$$

Example 30. *Consider the argument graph in Example 28. For this, $I_{cu}(G) = 1/3 + 2/3 + 3/4 = 7/4$.*

Proposition 9. *The I_{cu} measure is a graph-based inconsistency measure according to Definition 19.*

Proof. (Consistency) Assume $\mathsf{Arcs}(G) = \emptyset$. So $\sum_{(A_i,A_j)\in\mathsf{Arcs}(G)} \mathsf{Degree}(A_i, A_j)$ $= 0$. So $I_{cu}(G) = 0$. (Freeness) Assume $\mathsf{Nodes}(G) = \mathsf{Nodes}(G') \setminus \{A\}$ and $\mathsf{Arcs}(G) = \mathsf{Arcs}(G')$. So

$$\sum_{(A_i,A_j)\in\mathsf{Arcs}(G)} \mathsf{Degree}(A_i, A_j) = \sum_{(A_i,A_j)\in\mathsf{Arcs}(G')} \mathsf{Degree}(A_i, A_j)$$

Therefore, $I_{cu}(G) = I_{cu}(G')$. □

Proposition 10. *The I_{cu} measure satisfies Monotonicity, Inversion, Isomorphic invariance, Disjoint additivity, and Super-additivity.*

Proof. (Monotonicity) Assume $G \sqsubseteq G'$. So $\mathsf{Arcs}(G) \subseteq \mathsf{Arcs}(G')$. So,

$$\sum_{(A_i,A_j)\in\mathsf{Arcs}(G)} \mathsf{Degree}(A_i, A_j) \leq \sum_{(A_i,A_j)\in\mathsf{Arcs}(G')} \mathsf{Degree}(A_i, A_j)$$

So, $I_{cu}(G) \leq I_{cu}(G')$. (Inversion) Assume $G' = \mathsf{Invert}(G)$. Since Degree is symmetric,

$$\sum_{(A_i,A_j)\in\mathsf{Arcs}(G)} \mathsf{Degree}(A_i, A_j)$$
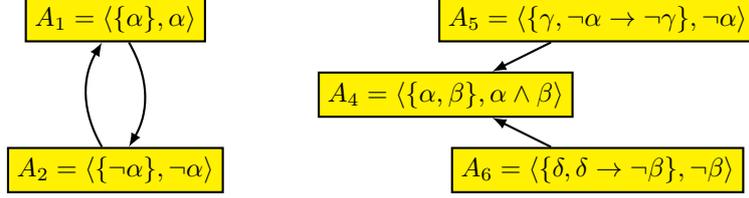$$= \sum_{(A_i,A_j)\in\mathsf{Arcs}(\mathsf{Invert}(G))} \mathsf{Degree}(A_j, A_i)$$

Therefore, $I_{cu}(G) = I_{cu}(\mathsf{Invert}(G))$. (Isomorphic invariance) Similar to proof for inversion. (Disjoint additivity) Assume G_1 and G_2 are disjoint. Therefore,

$$\sum_{(A_i,A_j)\in\mathsf{Arcs}(G)} \mathsf{Degree}(A_i, A_j) =$$
$$\sum_{(A_i,A_j)\in\mathsf{Arcs}(G_1)} \mathsf{Degree}(A_i, A_j)$$
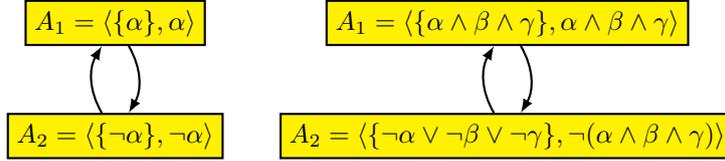$$+ \sum_{(A_i,A_j)\in\mathsf{Arcs}((G_2)} \mathsf{Degree}(A_j, A_i)$$

Therefore, $I_{cu}(G) = I_{cu}(G_1) + I_{cu}(G_2)$. (Super-additivity) Similar to proof for disjoint additivity. □

Proposition 11. *The I_{cu} measure is pairwise order-incomparable with each of the I_{dr}, I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc}, I_{ic}, I_{pr}, I_{ngr}, and I_{ust} measures.*

Proof. From the differences in satisfaction of properties in Proposition 3, 6, and 10, I_{cu} is pairwise incompatible with the I_{win}, I_{wou}, I_{ic}, I_{pr}, I_{ngr}, and I_{ust} measures. However, from the properties in Proposition 3, we cannot discriminate I_{cu} from I_{in}, I_{wcc}, and I_{cc}. To discriminate I_{cu} from I_{in}, consider the following graphs G_1 (left) and G_2 (right) where $I_{in}(G_1) = 2$ and $I_{in}(G_2) = 2$, whereas $I_{cu}(G_1) = 2$ and $I_{cu}(G_2) = 2/3$.

$A_1 = \langle \{\alpha\}, \alpha \rangle$

$A_2 = \langle \{\neg\alpha\}, \neg\alpha \rangle$

$A_5 = \langle \{\gamma, \neg\alpha \rightarrow \neg\gamma\}, \neg\alpha \rangle$

$A_4 = \langle \{\alpha, \beta\}, \alpha \wedge \beta \rangle$

$A_6 = \langle \{\delta, \delta \rightarrow \neg\beta\}, \neg\beta \rangle$

To discriminate I_{cu} from I_{cc}, consider the following graphs G_1 (left) and G_2 (right) where $I_{cc}(G_1) = 1$ and $I_{cc}(G_2) = 1$, whereas $I_{cu}(G_1) = 2$ and $I_{cu}(G_2) = 2/3$.

$A_1 = \langle \{\alpha\}, \alpha \rangle$

$A_2 = \langle \{\neg\alpha\}, \neg\alpha \rangle$

$A_1 = \langle \{\alpha \wedge \beta \wedge \gamma\}, \alpha \wedge \beta \wedge \gamma \rangle$

$A_2 = \langle \{\neg\alpha \vee \neg\beta \vee \neg\gamma\}, \neg(\alpha \wedge \beta \wedge \gamma) \rangle$

To discriminate I_{cu} from I_{wcc}, we can use a similar example to above. $\qquad \square$

In the proposals for degree of undercut [BH05, BH08], there are further options for degree of undercut, and these could be harnessed directly in the cumulative degree of undercut definition to provide potentially useful alternatives.

4.3 Application of logic-based measures of inconsistency

In this section, we harness logic-based inconsistency measures to measure inconsistency in an argument graph instantiated with deductive arguments. We start by reviewing a couple of simple logic-based inconsistency measures. The first is the number of minimal inconsistent subsets of the knowledgebase, and the second is the sum of the inverse of the cardinality of each minimal inconsistent subset.

Definition 32. *Let K be a set of propositional formulae, and let $\mathsf{MinIncon}(K)$ be the set of minimal inconsistent subsets of K. The I_M measure and the $I_\#$ measure are defined as follows.*

$$I_M(K) = |\mathsf{MinIncon}(K)| \qquad\qquad I_\#(K) = \sum_{X \in \mathsf{MinIncon}(K)} \frac{1}{|X|}$$

Example 31. *Let $K = \{\alpha, \neg\alpha \vee \neg\beta, \beta, \neg\gamma, \neg\gamma \rightarrow \neg\alpha\}$. So $\mathsf{MinIncon}(K)$ is as below, $I_M(K) = 2$ and $I_\#(K) = 2/3$.*
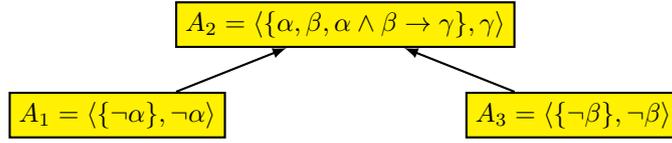
$$\mathsf{MinIncon}(K) = \{\{\alpha, \neg\alpha \vee \neg\beta, \beta\}, \{\alpha, \neg\gamma, \neg\gamma \rightarrow \neg\alpha\}\}$$

29

The cumulative attack inconsistency measure takes the sum of the inconsistency measure of the premises of each attacker and attackee.

Definition 33. *The* **cumulative attack inconsistency measure** *w.r.t logic-based inconsistency measure I' is*

$$I_{I'}^C(G) = \sum_{(A_i, A_j) \in \mathsf{Arcs}(G)} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j))$$

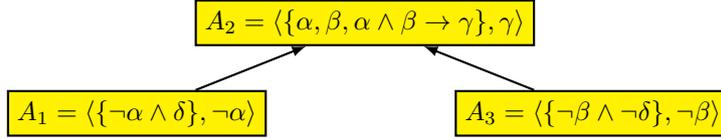**Example 32.** *For the following graph G, $I_{I_M}^C(G) = 2$ and $I_{I_\#}^C(G) = 1$.*

$$A_2 = \langle \{\alpha, \beta, \alpha \wedge \beta \to \gamma\}, \gamma \rangle$$

$$A_1 = \langle \{\neg\alpha\}, \neg\alpha \rangle \qquad A_3 = \langle \{\neg\beta\}, \neg\beta \rangle$$

The support inconsistency measure, defined next, takes the inconsistency measure of the premises of all the arguments in the graph taken together.

Definition 34. *The* **support inconsistency measure** *w.r.t logical inconsistency measure I' is*

$$I_{I'}^S(G) = I'(\bigcup_{A \in \mathsf{Nodes}(G)} \mathsf{Support}(A))$$

Example 33. *Continuing Example 32, $I_{I_M}^S(G) = 2$ and $I_{I_\#}^S(G) = 1$.*

Example 34. *For the following graph G, $I_{I_M}^C(G) = 2$, $I_{I_\#}^C(G) = 1$, $I_{I_M}^S(G) = 3$, and $I_{I_\#}^S(G) = 3/2$.*

$$A_2 = \langle \{\alpha, \beta, \alpha \wedge \beta \to \gamma\}, \gamma \rangle$$

$$A_1 = \langle \{\neg\alpha \wedge \delta\}, \neg\alpha \rangle \qquad A_3 = \langle \{\neg\beta \wedge \neg\delta\}, \neg\beta \rangle$$

Proposition 12. *The $I_{I'}^C$ measure is a graph-based inconsistency measure according to Definition 19.*

Proof. (Consistency) Assume $\mathsf{Arcs}(G) = \emptyset$. So $\sum_{(A_i, A_j) \in \mathsf{Arcs}(G)} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j)) = 0$. So $I_{I'}^C(G) = 0$. (Freeness) Assume $\mathsf{Nodes}(G) = \mathsf{Nodes}(G') \setminus \{A\}$ and $\mathsf{Arcs}(G) = \mathsf{Arcs}(G')$. So,

$$\sum_{(A_i, A_j) \in \mathsf{Arcs}(G)} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j))$$
$$= \sum_{(A_i, A_j) \in \mathsf{Arcs}(G')} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j))$$

So, $I_{I'}^C(G) = I_{I'}^C(G')$. $\qquad \square$

Definition 35. *Let G be an argument graph instantiated with deductive arguments. G is* **reflective** *iff if $\bigcup_{A \in \mathsf{Nodes}(G)} \mathsf{Support}(A) \vdash \bot$, then $\mathsf{Arcs}(G) \neq \emptyset$.*

Assumption 1. *For the rest of the paper, we assume that all the argument graphs are reflective.*

Despite having an intuitive rationale, $I_{I'}^S$ is not a graph-based inconsistency measure according to Definition 19.

Proposition 13. *The $I_{I'}^S$ measure satisfies consistency but not freeness (as given in Definition 19).*

Proof. (Consistency) Assume $\mathsf{Arcs}(G) = \emptyset$. Therefore, there are no arguments A_i and A_j such that A_i attacks A_j. Therefore, $\bigcup_{A \in \mathsf{Nodes}(G)} \mathsf{Support}(A) \nvdash \bot$. Therefore, $I_{I_M}^S(G) = 0$. (Freeness) Consider $A_1 = \langle \{\alpha \wedge \beta\}, \alpha \leftrightarrow \beta \rangle$, $A_2 = \langle \{\neg\alpha \wedge \gamma\}, \alpha \leftrightarrow \beta \rangle$, and $A_3 = \langle \{\neg\beta \vee \neg\gamma, \neg\beta \vee \neg\gamma \rightarrow \delta\}, \delta \rangle$. Let $\mathsf{Nodes}(G) = \{A_1, A_2\}$, $\mathsf{Arcs}(G) = \{(A_2, A_1)\}$, $\mathsf{Nodes}(G') = \{A_1, A_2, A_3\}$, and $\mathsf{Arcs}(G') = \{(A_2, A_1)\}$. So $I_{I_M}^S(G) = 1$ and $I_{I_M}^S(G') = 2$. $\qquad\square$

Assumption 2. *We assume for the rest of this paper that when an argument A appears in $\mathsf{Nodes}(G)$ and in $\mathsf{Nodes}(G')$, then the logical argument associated with the node is the same (i.e. $\mathsf{Support}(A)$ is the same in both graphs, and $\mathsf{Claim}(A)$ is the same in both graphs). In addition, for argument $A \in \mathsf{Nodes}(G)$ and $A' \in \mathsf{Nodes}(G')$, if $\mathsf{Support}(A) = \mathsf{Support}(A')$ and $\mathsf{Claim}(A) = \mathsf{Claim}(A')$ then A and A' have the same name (i.e. $A = A'$).*

Proposition 14. *The $I_{I'}^C$ measure satisfies Monotonicity, Inversion, Isomorphic invariance, Disjoint additivity, and Super-additivity.*

Proof. (Monotonicity) Assume $G \sqsubseteq G'$. So $\mathsf{Arcs}(G) \subseteq \mathsf{Arcs}(G')$. So,

$$\sum_{(A_i, A_j) \in \mathsf{Arcs}(G)} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j))$$
$$= \sum_{(A_i, A_j) \in \mathsf{Arcs}(G')} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j))$$

So, $I_{I'}(G) \leq I_{I'}(G')$. (Inversion) Assume $G' = \mathsf{Invert}(G)$. So,

$$\sum_{(A_i, A_j) \in \mathsf{Arcs}(G)} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j))$$
$$= \sum_{(A_i, A_j) \in \mathsf{Arcs}(\mathsf{Invert}(G))} I'(\mathsf{Support}(A_j) \cup \mathsf{Support}(A_i))$$

Therefore, $I_{I'}(G) = I_{I'}(\mathsf{Invert}(G))$. (Isomorphic invariance) Similar to proof for inversion. (Disjoint additivity) Assume G_1 and G_2 are disjoint. Therefore,

$$\sum_{(A_i, A_j) \in \mathsf{Arcs}(G)} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j)) =$$
$$\sum_{(A_i, A_j) \in \mathsf{Arcs}(G_1)} I'(\mathsf{Support}(A_i) \cup \mathsf{Support}(A_j))$$
$$+ \sum_{(A_i, A_j) \in \mathsf{Arcs}((G_2)} I'(\mathsf{Support}(A_j) \cup \mathsf{Support}(A_i))$$

Therefore, $I_{I'}(G) = I_{I'}(G_1) + I_{I'}(G_2)$. (Super-additivity) Similar to proof for disjoint additivity. $\qquad\square$

Proposition 15. *The $I_{I'}^S$ measure satisfies Monotonicity, Inversion, and Isomorphic invariance/ However, $I_{I'}^S$ does not satisfy Disjoint additivity, or Super-additivity.*
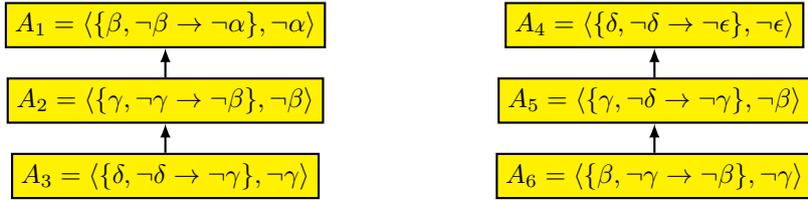
Proof. (Monotonicity) Assume $G \sqsubseteq G'$. So $\mathsf{Arcs}(G) \subseteq \mathsf{Arcs}(G')$. So

$$\bigcup_{A \in \mathsf{Nodes}(G)} \mathsf{Support}(A) \subseteq \bigcup_{A \in \mathsf{Nodes}(G')} \mathsf{Support}(A)$$

So, $I_{I'}^S(G) \leq I_{I'}^S(G')$. (Inversion) Assume $G' = \mathsf{Invert}(G)$. So,

$$\bigcup_{A \in \mathsf{Nodes}(G)} \mathsf{Support}(A) = \bigcup_{A \in \mathsf{Nodes}(G')} \mathsf{Support}(A)$$

Therefore, $I_{I'}^S(G) \leq I_{I'}^S(\mathsf{Invert}(G))$. (Isomorphic invariance) Similar to proof for inversion. (Disjoint additivity) Consider G_1 (left) and G_2 (right) where $I_{I_M}(G_1 + G_2) = 2\ I_{I_M}(G_1) = 2$ and $I_{I_M}(G_2) = 2$.

(Super-additivity) Similar to proof for disjoint additivity. □

For pairwise order-incompatibility, we consider $I' = I_M$ below. We can obtain similar results for other instantiations of $I_{I'}^C$ and $I_{I'}^S$.

Proposition 16. *The $I_{I_M}^C$ and $I_{I_M}^S$ measures are pairwise order-incompatible with each of the I_{dr}, I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc}, I_{ic}, I_{pr}, I_{ngr}, I_{ust}, and I_{cu} measures.*

Proof. From the differences in satisfaction of properties in Propositions 3, 6, and 10, $I_{I_M}^C$ is pairwise incompatible with the I_{win}, I_{wou}, I_{ic}, I_{pr}, I_{ngr}, and I_{ust} measures. However, from the properties in Proposition 3, we cannot discriminate $I_{I_M}^C$ from I_{in}, I_{cc}, I_{wcc}, and I_{cu}. To discriminate $I_{I_M}^C$ from I_{in}, consider the following graphs G_1 (left) and G_2 (right) where $I_{in}(G_1) = 2$ and $I_{in}(G_2) = 1$, whereas $I_{I_M}^C(G_1) = 2$ and $I_{I_M}^C(G_2) = 2$.

To discriminate $I^C_{I_M}$ from I_{cc}, consider the above graphs G_1 (above left) and G_2 (above right) where $I_{cc}(G_1) = 1$ and $I_{cc}(G_2) = 0$, whereas $I^C_{I_M}(G_1) = 2$ and $I^C_{I_M}(G_2) = 2$. To discriminate $I^C_{I_M}$ from I_{wcc} and from I_{cu}, we can use a similar example to above. To show the $I^S_{I_M}$ measure is pairwise incompatible with each of the I_{dr}, I_{in}, I_{win}, I_{wou}, I_{cc}, I_{wcc}, I_{ic}, I_{pr}, I_{ngr}, I_{ust}, and I_{cu} measures, we can use the failure of freeness to create examples where order-compatibility fails for each pairwise comparison. $\square$

In this subsection, we have harnessed two existing logic-based inconsistency measures, I_M and $I_\#$, for measuring inconsistency in argument graphs instantiated with deductive arguments. There is a wide range of further measures of inconsistency that we could deploy in this role (for reviews see [GH11b, Thi16]). In addition, we have only considered two ways of applying logic-based measures, namely $I^C_{I'}$ and $I^S_{I'}$. Further, ways of applying logic-based measures include analyzing the support in extensions of an argument graph to identify inconsistency. For instance, it is not necessarily the case that the union of the support of the arguments in an extension is consistent (for more discussion of this point, see [GH11a]). Another option is to check the inconsistency measure of the premises of the defenders of an argument since it is not necessarily the case that these would be consistent. We leave investigation of these options to further work.

4.4   Related work

In the converse of what we have considered in this section, deductive argumentation has been used for measuring inconsistency. For this, an argument tree (as defined by [BH01, BH08]) is used. Each node in the tree is an argument. Each child is a canonical undercut of its parents. For each node in the tree, each canonical undercut of the node is a child of the node (except when the premises of the child have all occurred in the support of the argument in ancestor arguments). This exception prohibits infinite branches where each argument has the same premises that have already occurred on the branch.

Example 35. *Consider the knowledgebase $K = \{\alpha, \neg\alpha, \neg\alpha \vee \beta, \beta, \neg\beta\}$. The following is an argument tree with A_1 being the root.*
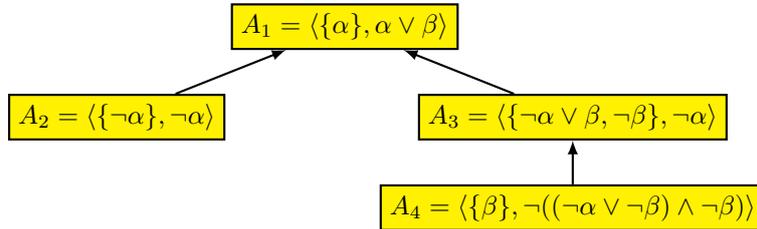
In [Rad15], an argument tree is constructed where the argument at the root has a single premise. Then, three proposals are made for evaluating the inconsistency of this formula.

Definition 36. *Let T be the argument tree with the argument A at the root having* $\mathsf{Support}(A) = \phi$. *An* I_{ARG}^x *measure is defined as follows*

$$I_{ARG}^x(\phi, T) = |\mathsf{Undercuts}(\phi, T)| \times f^x(\phi, T)$$

where $\mathsf{Undercuts}(\phi, T)$ *is the set of undercuts of the root argument in the tree* T, $\mathsf{Height}(T)$ *is the height of the tree (i.e. the number of edges in a path from root to leaf on the longest branch), and* $\mathsf{Depth}(n, T)$ *is the depth of node n in the tree (i.e. the number of edges in a path from root to node n).*

$$f^1(\phi, T) = \frac{1}{\mathsf{Height}(T)}$$

$$f^2(\phi, T) = \frac{1}{\sum_{n \in \mathsf{Nodes}(T) \setminus \{\phi\}} \mathsf{Depth}(n, T)}$$

$$f^3(\phi, T) = \sum_{n \in \mathsf{Nodes}(T) \setminus \{\phi\}} \frac{1}{\mathsf{Depth}(n, T)}$$

So $I_{ARG}^1(\phi, T)$ takes the height of the tree into account, $I_{ARG}^2(\phi, T)$ takes the inverse of the sum of the depth of each node into account, and $I_{ARG}^3(\phi, T)$ takes the sum of the inverse of the depth of each node into account.

Example 36. *Continuing Example 35, the measures for three of the formulae are tabulated.*

	α	$\neg\alpha \vee \beta$	$\neg\alpha$
I_{ARG}^1	1	1/2	1/3
I_{ARG}^2	1/2	1/5	1/6
I_{ARG}^3	5	2	11/6

Whilst the proposal by Raddaoui [Rad15] is for measuring inconsistency in a formula, it is possible that the ideas could be adapted for measuring the inconsistency of argument graphs.

5 Resolution through commitment

An agent can commit to some arguments (i.e. declare whether they think an argument is acceptable or not), and s/he can be queried about those commitments. We assume that commitment by an agent is represented by the belief an agent has in the arguments.

In this section, we consider how we can model the agent. For this, we assume a labelling function as defined in Section 2.2. Initially, if we know nothing about the agent, we start with a uniform labelling that assigns undec to each argument. Then suppose the agent declares a commitment to an argument — either by saying that the label for the argument is in or that it is out — we can consider what the ramifications are of that commitment on the other beliefs, and moreover, we can use it for resolving inconsistency (i.e. reducing the measure of inconsistency of the argument graph).

We proceed by introducing some subsidiary definitions for labellings, and for generating a subgraph of a graph based on a labelling. The first subsidiary definition specifies a labelling for which there is no undecided label.

Definition 37. *A labelling L is **committed** for graph G iff for all $A \in$ Nodes(G), $A \in$ in(G) or $A \in$ out(G).*

Example 37. *Consider the following graph. For this, the following labelling is committed: $L(A1) =$ out, $L(A2) =$ in, and $L(A3) =$ out.*
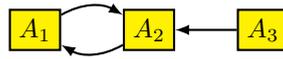
The next subsidiary definition constrains a labelling to take into account the attack relationship. As we illustrate in the subsequent example, a strict labelling is not necessarily an admissible labelling, though every admissible labelling is a strict labelling.

Definition 38. *A labelling L is **strict** for graph G iff for all $(A, B) \in$ Arcs(G), if $A \in$ in(G), then $B \in$ out(G).*

Example 38. *Consider the graph in Example 37. For this, the labellings that are committed and strict are tabulated below.*

	A_1	A_2	A_3
L_1	out	out	out
L_2	in	out	out
L_3	out	out	in
L_4	in	out	in
L_5	out	in	out

The following definition forms a subgraph from a graph and a labelling by deleting every node that is labelled out and deleting every arc that has either the source or the target labelled out. The reason we want this new graph is that if an agent commits to an argument being out, then that argument is no longer acceptable, and we can ignore it from further consideration.
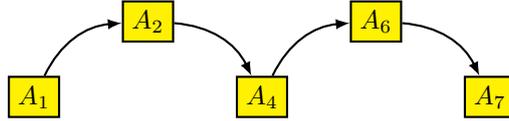
Definition 39. *Given an argument graph G and labelling L, the **new graph** function is NewGraph$(G, L) = G'$ where*

35

- $\mathsf{Nodes}(G') = \{A \in \mathsf{Nodes}(G) \mid L(A) \neq \mathsf{out}\}$

- $\mathsf{Arcs}(G') = \{(A, B) \in \mathsf{Arcs}(G) \mid L(A) \neq \mathsf{out} \text{ and } L(B) \neq \mathsf{out}\}$

Example 39. *Consider the following graph G with the labelling $L(A_1) = $ undec, $L(A_2) = $ undec, $L(A_3) = $ out, $L(A_4) = $ in, $L(A_5) = $ out, $L(A_6) = $ in, and $L(A_7) = $ undec.*
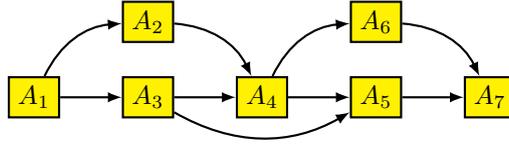
*So the new graph G' for this graph and labelling is below.*

We can apply a measure of inconsistency to the graph G and G', and determine the reduction in inconsistency. For instance, $I_{in}(G) = 9$ and $I_{in}(G') = 4$. Similarly, $I_{win}(G) = 9/2$ and $I_{win}(G') = 4$.

When we query an agent about an argument A, we get a reply of either in or out. Given this information, we need to update the labelling to L' as follows.

Definition 40. *Let L be a labelling for graph G, and let $A \in \mathsf{Nodes}(G)$ be a query.*

- *If the answer for A is in, then*
  - *$L'(A) = $ in*
  - *for each $(A, B) \in \mathsf{Nodes}(G)$, $L'(B) = $ out,*
  - *for each $(B, A) \in \mathsf{Nodes}(G)$, $L'(B) = $ out,*
  - *for all other arguments C, $L'(C) = L(C)$.*

- *If the answer for A is out, then*
  - *$L'(A) = $ out*
  - *for all other arguments C, $L'(C) = L(C)$.*

Example 40. *Consider the following graph. Let L_1 be the original labelling, let L_2 be the new labelling obtained after the first query, and let L_3 be the new labelling obtained after the second query.*
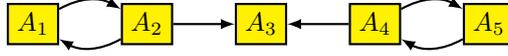
Suppose the first query concerns A_1, with the reply out, and the second query concerns A_4, with the reply in. The labellings are tabulated below.

	A_1	A_2	A_3	A_4	A_5
L_1	undec	undec	undec	undec	undec
L_2	out	undec	undec	undec	undec
L_3	out	undec	out	in	out

Now we can show how measures of inconsistency can help in deciding which arguments to query. We illustrate this in the following example.

Example 41. *Consider the following graph G where $I_{in}(G) = 6$ and $I_{cc}(G) = 2$. Suppose $L_0(A_i) = $ undec for all $A_i \in \{A_1, \ldots, A_5\}$. We could query any of these arguments.*
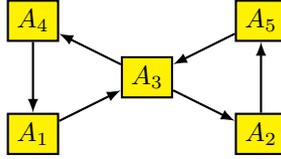
In the following, each bullet point concerns a specific query and specific answer to that query. In each case, given a graph G, and the revised labelling L, we obtain the new graph G' where $\mathsf{NewGraph}(G, L) = G'$.

- *If we query A_3, and we get the answer A_3 is in, then the resulting labelling is*

$$L(A_1) = \mathsf{out} \quad L(A_2) = \mathsf{out} \quad L(A_3) = \mathsf{in}$$
$$L(A_4) = \mathsf{out} \quad L(A_5) = \mathsf{out}$$

  - *Hence, $\mathsf{NewGraph}(G, L) = (\{A_3\}, \{\})$.*
  - *So $I_{in}(G') = 0$ and $I_{cc}(G') = 0$.*

- *If we query A_3, and we get the answer A_3 is out, then the resulting labelling is*

$$L(A_1) = \mathsf{undec} \quad L(A_2) = \mathsf{undec} \quad L(A_3) = \mathsf{out}$$
$$L(A_4) = \mathsf{undec} \quad L(A_5) = \mathsf{undec}$$

  - *Hence, $\mathsf{NewGraph}(G, L) = (\{A_1, A_2, A_4, A_5\}, \{(A_4, A_1), (A_2, A_5)\})$.*
  - *So $I_{in}(G') = 2$ and $I_{cc}(G') = 0$.*

- *If we query A_1, and we get the answer A_1 is in, then the resulting labelling is*

$$L(A_1) = \text{in} \quad L(A_2) = \text{undec} \quad L(A_3) = \text{out}$$
$$L(A_4) = \text{out} \quad L(A_5) = \text{undec}$$

  - *Hence, $\text{NewGraph}(G, L) = (\{A_1, A_2, A_5\}, \{(A_2, A_5)\})$.*
  - *So $I_{in}(G') = 1$ and $I_{cc}(G') = 0$.*

- *If we query A_1, and we get the answer A_1 is out, then the resulting labelling is*

$$L(A_1) = \text{out} \quad L(A_2) = \text{undec} \quad L(A_3) = \text{undec}$$
$$L(A_4) = \text{undec} \quad L(A_5) = \text{undec}$$

  - *Hence, $\text{NewGraph}(G, L) =$*

$$(\{A_2, A_3, A_4, A_5\}, \{(A_3, A_4), (A_2, A_5), (A_3, A_2), (A_5, A_3)\})$$

  - *So $I_{in}(G') = 4$ and $I_{cc}(G') = 1$.*

- *If we query A_2, and we get the answer A_2 is in, then the resulting labelling is*

$$L(A_1) = \text{undec} \quad L(A_2) = \text{in} \quad L(A_3) = \text{out}$$
$$L(A_4) = \text{undec} \quad L(A_5) = \text{out}$$

  - *Hence, $\text{NewGraph}(G, L) = (\{A_1, A_2, A_4\}, \{(A_4, A_1)\})$.*
  - *So $I_{in}(G') = 1$ and $I_{cc}(G') = 0$.*

- *If we query A_2, and we get the answer A_2 is out, then the resulting labelling is*

$$L(A_1) = \text{undec} \quad L(A_2) = \text{out} \quad L(A_3) = \text{undec}$$
$$L(A_4) = \text{undec} \quad L(A_5) = \text{undec}$$

  - *Hence, $\text{NewGraph}(G, L) =$*

$$(\{A_1, A_3, A_4 A_5\}, \{(A_1, A_3), (A_3, A_4), (A_4, A_1), (A_5, A_3)\})$$

  - *So $I_{in}(G') = 4$ and $I_{cc}(G') = 1$.*

Because of the graph structure, querying A4 is the same as querying A2 and querying A5 is the same as querying A1. Therefore, if we query A_3, we will have the maximum reduction in inconsistency if we use the I_{in} or I_{cc} measures where the reduction is the average of the reduction for the in and out answers.

In this section, we have seen how we can use the inconsistency measures for graphs as a way of guiding the selecting of arguments to query an agent. The answer from the agent is a commitment to accepting or not accepting the argument, and this can be used to resolve inconsistencies in the graph.

6 Discussion

This chapter makes the following contributions: (1) A proposal for a general framework of postulates for characterizing measures of inconsistency for argument graphs; (2) Proposals for graph structure measures and graph extension measures as instances of measures of inconsistency for argument graphs; (3) A review of the degree of undercut approach to measuring inconsistency for argument graphs instantiated with deductive arguments; (4) An investigation of the use of existing (logic-based) measures of inconsistency to measuring inconsistency for argument graphs instantiated with deductive arguments; and (5) An outline of how measures of inconsistency for argument graphs can be used as part of a process for inconsistency resolution in argumentation.

In future work, it would be good to give further properties for an inconsistency measure and relationships between them, further definitions for an inconsistency measure, further results for specific classes of graphs, and methods for resolution based on commitments.

The structure-based measures are in a sense measuring aspects of graph complexity. There are numerous options for features of argument graphs that could be considered (see for instance, features used for selecting argument solvers [VCG14]). Further options for measuring graph complexity include measuring the sparseness of the graph (e.g. average indegree, average outdegree, indegree distribution, or outdegree distribution) which may give more recognition to unattacked arguments, radius of the graph (which is the maximum eccentricity of any node in the graph where the eccentricity of a node is the length of the shortest path to the node furthest away from that node), and dimensions of the graph (i.e. the minimum number of dimensions of Euclidean space required to represent all arcs with unit length). Another possible field for options for graph-based measures of inconsistency is graph entropy (see for example [DM11]).

A comparison with proposals for argument strength would be interesting. These consider a weight to individual arguments which can be affected by arguments that impinge upon it. A number of proposals have been made (e.g. [BH01, CLS05, MT08, ABN13, ABN15, ABN16, ABNDV16, GM15, TKI14, BDKM16a]), and comparisons undertake with a range of postulates (for a review see [BDKM16b]). Possibly, analogous postulates can be proposed for inconsistency measures in graphs. Also it would be interesting to consider relationships with approaches to weighted labeling (e.g. [dDTV16]).

It would also be interesting to consider inconsistency measures for other forms of structured argumentation such as assumption-based argumentation (for a tutorial, see [Ton14]), defeasible logic programming (for a tutorial, see [GS14]), and ASPIC+ (for a tutorial, see [MP14]). This would involve developing logic-based inconsistency measures for these non-standard logical formalisms.

Finally, it would be valuable to apply these techniques as part of the process for inconsistency resolution in argumentation for an application such as intelligence analysis to investigate the usability of the measures, and whether indeed there are tangible benefits to using these inconsistency measures.

Acknowledgments

References

[ABG+17]   K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. Simari, M. Thimm, and S. Villata. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, 2017.

[ABN13]   L. Amgoud and J. Ben-Naim. Ranking-based semantics for argumentation frameworks. In *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM'13)*, volume 8078 of *Lecture Notes In Artificial Intelligence*, pages 134–147, Washington, DC, USA, September 2013.

[ABN15]   L. Amgoud and J. Ben-Naim. Argumentation-based ranking logics. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, pages 1511–1519, 2015.

[ABN16]   L. Amgoud and J. Ben-Naim. Axiomatic foundations of acceptability semantics. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, 2016.

[ABNDV16]   L. Amgoud, J. Ben-Naim, D. Doder, and S. Vesic. Ranking arguments with compensation-based semantics. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, pages 12–21, 2016.

[BCD07]   T. Bench-Capon and P. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[BDKM16a]   E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet. Argumentation ranking semantics based on propagation. In *Proceedings of*

the Sixth International Conference on Computational Models of Argumentation (COMMA'16), pages 139–150, 2016.

[BDKM16b] E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet. A comparative study of ranking-based semantics for abstract argumentation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 914–920, 2016.

[Bes14]   Ph. Besnard. Revisiting postulates for inconsistency measures. In *Logics in Artificial Intelligence (JELIA'14)*, LNCS, pages 383–396. Springer, 2014.

[BGGv18]   P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, editors. *Handbook of Formal Argumentation*. College Publications, 2018. in preparation.

[BH01]   Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.

[BH05]   Ph. Besnard and A. Hunter. Practical first-order argumentation. In *Proceedings of the 20th American National Conference on Artificial Intelligence (AAAI'05)*, pages 590–595. MIT Press, 2005.

[BH08]   Ph. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008.

[BH14]   Ph. Besnard and A. Hunter. Constructing argument graphs with deductive arguments: A tutorial. *Argument and Computation*, 5(1):5–30, 2014.

[Cay95]   C. Cayrol. On the relation between argumentation and non-monotonic coherence-based entailment. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1443–1448, 1995.

[CG09]   M. Caminada and D. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[CLS05]   C. Cayrol and M.-C. Lagasquie-Schiex. Graduality in argumentation. *Journal of Artificial Intelligence Research*, 23:245–297, 2005.

[Dal88]   M. Dalal. Investigations into a theory of knowledge base revision. In *Proceedings of National Conference on Artificial Intelligence (AAAI'88)*, pages 475–479, 1988.

[dDTV16]   C. da Costa Pereira, M. Dragoni, A. Tettamanzi, and S. Villata. Fuzzy labeling for abstract argumentation: An empirical evaluation. In *Proceedings of the 10th International Conference on Scalable Uncertainty Management (SUM'13)*, volume 9858 of *Lecture Notes In Artificial Intelligence*, pages 126–139, 2016.

[DHM+11]   P. E. Dunne, A. Hunter, P. McBurney, S. Parsons, and M. Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175:457–486, 2011.

[DM11]   M. Dehmer and A. Mowshowitz. A history of graph entropy measures. *Information Sciences*, 181:5778, 2011.

[DRMO10]   D. Doder, M. Rašković, Z. Marković, and Z. Ognjanović. Measures of inconsistency and defaults. *International Journal of Approximate Reasoning*, 51(7):832–845, 2010.

[Dun95]   P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[GH11a]   N. Gorogiannis and A. Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence*, 175(9-10):1479–1497, 2011.

[GH11b]   J. Grant and A. Hunter. Measuring the good and the bad in inconsistent information. In *International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 2632–2637, 2011.

[GM15]   D. Grossi and S. Modgil. On the graded acceptability of arguments. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*, pages 868–874, 2015.

[GS14]   A. Garcia and G. Simari. Defeasible logic programming: Delp-servers, contextual queries, and explanations for answers. *Argument and Computation*, 5(1):63–88, 2014.

[HK04]   A. Hunter and S. Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, volume 3300 of *LNCS*, pages 189–234. Springer, 2004.

[HK10]   A. Hunter and S. Konieczny. On the measure of conflicts: Shapley inconsistency values. *Artificial Intelligence*, 174:1007–1026, 2010.

[JMR14]   S. Jabbour, Y. Ma, and B. Raddaoui. Inconsistency measurement thanks to mus decomposition. In *Proceedings of the 2014 international conference onAutonomous agents and multi-agent systems*, pages 877–884. International Foundation for Autonomous Agents andMultiagent Systems, 2014.

[Kni01]   K. Knight. Measuring inconsistency. *Journal of Philosophical Logic*, 31:77–98, 2001.

[MLJ12]   K. Mu, W. Liu, and Z. Jin. Measuring the blame of each formula for inconsistent prioritized knowledge bases. *Journal of Logic and Computation*, 22(3):481–516, 2012.

[MP14]   S. Modgil and H. Prakken. The ASPIC+ framework for structured argumentation. *Argumentation and Computation*, 5(1):31–62, 2014.

[MT08]   P.-A. Matt and F. Toni. A game-theoretic perspective on the notion of argument strength in abstract argumentation. In *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08)*, pages 285–297. Springer Berlin Heidelberg, 2008.

[Rad15]   B. Raddaoui. Computing inconsistency using logical argumentation. In *Proceedings of the International Conference on Agents and Artificial Intelligence ICAART*, pages 164–172, 2015.

[RS09]   I. Rahwan and G. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, 2009.

[Thi16]   M. Thimm. On the expressivity of inconsistency measures. *Artificial Intelligence*, 234:120–151, 2016.

[TKI14]   M. Thimm and G. Kern-Isberner. On controversiality of arguments and stratified labelings. In *Proceedings of the Fifth International Conference on Computational Models of Argumentation (COMMA'14)*, pages 413–420, September 2014.

[TNE+15]   A. Toniolo, T. Norman, A. Etuk, F. Cerutti, R. Ouyang, M. Srivastava, N. Oren, T. Dropps, J. Allen, and P. Sullivan. Agent support to reasoning with different types of evidence in intelligence analysis. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 781–789, 2015.

[Ton14]   F. Toni. A tutorial on assumption-based argumentation. *Argumentation and Computation*, 5(1):89–117, 2014.

[VCG14]   M. Vallati, F. Cerutti, and M. Giacomin. Argumentation frame-
           works features: an initial study. In *Proceedings of the European
           Conference on Artificial Intelligence (ECAI'14)*, pages 1117 –
           1118, 2014.