FernUniversität in Hagen
Fakultät für Mathematik und Informatik

AIG
Artificial Intelligence Group

Characteristics of Inconsistency Measures in Argument Graphs in Abstract Argumentation

Bachelorarbeit

zur Erlangung des Grades einer Bachelor of Science (B.Sc.)
im Studiengang Informatik

vorgelegt von
Isabel Müßig

Erstgutachter: Prof. Dr. Matthias Thimm
Artificial Intelligence Group

Betreuer: Prof. Dr. Matthias Thimm
Artificial Intelligence Group

Erklärung

Ich erkläre, dass ich die Bachelorarbeit selbstständig und ohne unzulässige Inanspruchnahme Dritter verfasst habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht. Die Versicherung selbstständiger Arbeit gilt auch für enthaltene Zeichnungen, Skizzen oder graphische Darstellungen. Die Bachelorarbeit wurde bisher in gleicher oder ähnlicher Form weder derselben noch einer anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht. Mit der Abgabe der elektronischen Fassung der endgültigen Version der Bachelorarbeit nehme ich zur Kenntnis, dass diese mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate geprüft werden kann und ausschließlich für Prüfungszwecke gespeichert wird.

Der Veröffentlichung dieser Arbeit auf der Webseite des Lehrgebiets Künstliche Intelligenz und damit dem freien Zugang zu dieser Arbeit stimme ich ausdrücklich zu.

Für diese Arbeit erstellte Software wurde quelloffen verfügbar gemacht, ein entsprechender Link zu den Quellen ist in dieser Arbeit enthalten. Gleiches gilt für angefallene Forschungsdaten.

Abu Dhabi, 17.03.2025 .... Müßig

(Ort, Datum) (Unterschrift)

Zusammenfassung

Inkonsistenzmaße in abstrakten Argumentationsgraphen dienen der Quantifizierung der Inkonsistenz, die in Graphen vorhanden ist, welche die Beziehungen zwischen Argumenten darstellen. Zur Bewertung der Eigenschaften von Inkonsistenzmaßen gibt es verschiedene Ansätze, darunter die Prüfung der Konformität mit Rationalitätspostulaten, die Analyse ihrer Expressivität, die Berechnung ihrer Rechenkomplexität sowie deskriptive Bewertungen. Ziel dieser Bachelorarbeit ist es, einen umfassenden Überblick über den aktuellen Stand der Inkonsistenzmaße in abstrakten Argumentationsgraphen zu geben und gleichzeitig eine tiefere, bisher unerforschte Perspektive durch die Untersuchung der Eigenschaften der Inkonsistenzmaße auf der Ebene von Rationalitätspostulaten und deskriptiven Bewertungen zu bieten. Um dies zu erreichen, werden die notwendigen Präliminarien vorgestellt, gefolgt von einer gründlichen Analyse der Inkonsistenzmaße, wobei dessen Stärken und Schwächen hervorgehoben werden. Anschließend werden 18 Rationalitätspostulate für abstrakte Argumentationsgraphen untersucht, von denen 6 neu formuliert worden sind und auf etablierten Konzepten der klassischen Logik aufbauen. Abschließend werden die Konformität der Inkonsistenzmaße mit den Rationalitätspostulaten überprüft, und dessen Ergebnisse diskutiert. Obwohl die Ergebnisse je nach Schwerpunkt der Inkonsistenzmaßnahmen variieren, sticht das Distance-based Measure, als sehr feingranulares Maß hervor.

Abstract

Inconsistency measurements in abstract argument graphs are concerned with quantifying the inconsistency that exists in graphs which represent the relationships between arguments. To evaluate the properties of inconsistency measurements various approaches have been proposed including the examination of compliance with rationality postulates, the analysis of their expressivity characteristics, the calculation of their computational complexities as well as descriptive evaluations. The aim of this Bachelor thesis is, to present a comprehensive overview of the current state of inconsistency measures in abstract argument graphs, while offering a deeper, previously unexplored perspective through an investigation of their characteristics along the dimension of rationality postulates and descriptive evaluations. To achieve this, the necessary preliminaries are presented, followed by a thorough analysis and a detailed breakdown of each inconsistency measure, highlighting its strengths and weaknesses. Afterwards, 18 rationality postulates for abstract argument graphs, 6 of which are newly formulated, while building upon established concepts in classical logic, are examined. Finally, we check for compliance of the inconsistency measures with the rationality postulates and discuss those findings. Even though results vary based on the focus of the inconsistency measures, the Distance-based Measure stands out as a very fine-granular measure.

Contents

1 Introduction **1**

2 Preliminaries **3**
2.1 Argument Graphs . . . 3
2.2 Subsidiary Definitions . . . 4
2.3 Semantics . . . 6
2.4 Standard and Strong Equivalence . . . 8

3 Inconsistency Measures **11**
3.1 Graph Extension Measures . . . 11
3.1.1 PreferredCount . . . 12
3.1.2 NonGroundedCount . . . 14
3.1.3 UnstableCount . . . 17
3.1.4 NaiveExtensionCount . . . 19
3.1.5 PreferredExtensionCount . . . 20
3.2 Graph Structure Measures . . . 22
3.2.1 Drastic . . . 22
3.2.2 WeightedInSum . . . 23
3.2.3 WeightedOutSum . . . 24
3.2.4 CycleCount . . . 25
3.2.5 WeightedCycleCount . . . 26
3.2.6 WeightedComponentCount . . . 29
3.2.7 Connectance Measure . . . 30
3.2.8 In-degree Measure . . . 31
3.2.9 Distance-based Measure . . . 32

4 Analysis **35**
4.1 Rationality Postulates . . . 37
4.1.1 Basic Postulates . . . 37
4.1.2 Expansion Postulates . . . 38
4.1.3 Structural Postulates . . . 39
4.1.4 Strong Equivalence Postulates . . . 40
4.1.5 Additivity Postulates . . . 40
4.1.6 Cyclicality Postulates . . . 41
4.2 Compliance of Inconsistency Measures with Rationality Postulates . 43
4.2.1 Basic Postulate Propositions and Proofs . . . 45
4.2.2 Expansion Postulate Propositions and Proofs . . . 51
4.2.3 Strong Equivalence Postulate Propositions and Proofs . . . 57
4.2.4 Additivity Postulate Propositions and Proofs . . . 69
4.2.5 Cyclicaltiy Postulate Propositions and Proofs . . . 73

5 Conclusion **78**
5.1 Summary of Results 78
5.2 Future Work 80

List of Figures

1 Argument graph A . . . 1
2 Argument graphs B, C, and D . . . 4
3 Argument graphs C', E, F, and H . . . 5
4 Argument graphs J, $J_{\cong}$, and J' . . . 5
5 Argument graph K . . . 6
6 Implications between semantics . . . 7
7 Argument graphs L, L', and L'' . . . 10
8 Argument graphs M, N, and O . . . 12
9 Argument graphs P and its induced subgraphs . . . 17
10 Argument graphs Q, R, and R^{-1} . . . 23
11 Argument graph S . . . 28
12 Argument graph T . . . 42
13 Argument graph T' . . . 42
14 Argument graphs U and V . . . 50
15 Argument graphs L_1 and L'_1 . . . 59
16 Argument graphs L_2 and L'_2 . . . 60
17 Argument graphs L_3, L_4, $L_{3,4}^{g\mathcal{K}}$, L_5, and $L_5^{g\mathcal{K}}$ . . . 62
18 Argument graphs L_6 and L'_6 . . . 62
19 Argument graphs L_7 and L'_7 . . . 64
20 Argument graphs L_8 and L'_8 . . . 65
21 Argument graphs L_9 and L'_9 . . . 66
22 Argument graph $A + V$ . . . 70
23 Argument graphs W, Y, and $W + Y$ . . . 71
24 Argument graphs Z and $A + Z$ . . . 72

List of Tables

1 Semantics of argument graph B 8
2 Semantics of argument graph F 8
3 Semantics of argument graph L' 9
4 Semantics of argument graph J 9
5 Semantics of argument graph J' 9
6 Semantics of argument graph M 12
7 Semantics of argument graph N 13
8 Semantics of argument graph O 13
9 Semantics of argument graph E 14
10 Semantics of argument graph A 15
11 Semantics of argument graph $M + A$ 16
12 Semantics of argument graph H 16
13 Semantics of argument graph P 18
14 Overview of rationality postulates and their sets 36
15 Satisfaction and violation of the inconsistency measure postulates . . 43
16 Semantics of argument graph D 51
17 Semantics of argument graph L 54
18 Semantics of argument graph L_6 63
19 Semantics of argument graph L'_6 63
20 Semantics of argument graph L_5 63
21 Semantics of argument graph L_8 65
22 Semantics of argument graph L'_8 66
23 Semantics of argument graph $M + C'$ 68
24 Semantics of argument graph $A + V$ 70
25 Semantics of argument graph W 71
26 Semantics of argument graph Y 71
27 Semantics of argument graph $W + Y$ 72
28 Semantics of argument graph T'_4 74

1 Introduction

Argumentation is defined as the interaction of different arguments for or against some conclusion [47], where an argument is an entity that represents a position, claim, or assertion [46]. As an example, consider the following arguments which are assigned the labels a and b.

a = "Wind turbines should be installed in coastal regions to maximize wind energy."
b = "Wind turbines should not be installed in coastal regions because they disrupt the coastal ecosystem and tourism."

One of the ways in which arguments can be represented is through the utilization of an abstract argumentation framework or argument graph [17]. An argument graph consists of nodes, where each node represents an individual argument. The connections between nodes, called arcs, denote when one argument is in conflict with another argument or even with itself [10]. The arcs are directional to show which argument is attacking which [18]. Figure 1 shows the arguments a and b from above, pictured as an argument graph where a attacks b, and b attacks a.

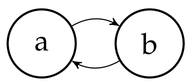


Figure 1: Argument graph A.

In argument graph A, conflict is represented through a attacking b, and b attacking a. Each argument disputes the validity of the other by disagreeing with each other on the installation of wind turbines with opposing reasons. Conflicts such as the one described for argument graph A, contribute to inconsistency.

Inconsistency refers to the fact that it is not possible to accept all the arguments simultaneously without running into contradictions [40]. In graph A all arguments cannot hold simultaneously - the graph is inconsistent. Beyond just identifying inconsistency, it is possible to measure the degree to which it occurs through inconsistency measures which quantify the degree of inconsistency in an argument graph [40].

Argument graphs and the quantification of inconsistency in them are relevant in the field of artificial intelligence (AI) [8]. By analyzing argument graphs for inconsistencies, AI systems can detect conflicting information and assess the strength and validity of different arguments [33]. This leads to more reliable AI systems that can handle conflicting data [2]. Additionally, the characteristics of inconsistency measures are crucial where characteristics refer to their features or properties.

The characteristics of inconsistency measures can be examined formally from many different dimensions, through rationality postulates [38], expressivity [37], computational complexity [39], but also qualitatively through descriptive evaluation [1]. They influence the reliability and robustness of AI systems because they determine how inconsistency is quantified and addressed. Moreover, each feature of an inconsistency measure is specifically designed to address distinct challenges and requirements encountered by AI systems, enabling it to be adapted to a specific context and goals of the application. For instance, decision-making in the medical field may require fine-grained inconsistency detection, while a robotic control system may prioritize a faster inconsistency detection. In its entirety, the characteristics of inconsistency measures determine how well the system can manage conflicts and adapt to specific decision-making contexts.

The goal of this thesis is to present a comprehensive overview of the current state of inconsistency measures in argument graphs within abstract argumentation, while offering a deeper, previously unexplored perspective through an examination along the dimension of rationality postulates and descriptive evaluation. Specifically, this thesis aims to answer the question what characteristics the inconsistency measures in argument graphs in abstract argumentation fulfill. The investigation of characteristics is done by determining what rationality postulates of inconsistency defined by [24], [1], [39], and [40] including six novel principles, Penalty, Free-Node Dilution, p-Exchange, g-Exchange, s-Exchange, and $naive$-Exchange, inspired by [35], [28] as well as, [9], [30], and [19], respectively, the measures fulfill. Additionally, the individual strengths and weaknesses of the measures are considered as descriptive evaluation. Due to the absence of existing papers on the strengths and weaknesses of the measures, and the diversity of the measures, the methodology will involve a detailed deconstruction of the measures, with conclusions drawn based on this analysis.

To summarize, this thesis contributes by examining inconsistency measures in argument graphs, analyzing their properties through rationality postulates, including six newly introduced ones, as well as by considering their strengths and weaknesses. The results of this thesis therefore include a classification of inconsistency measures based on their adherence to rationality postulates, highlighting which measures satisfy which principles. Additionally, the descriptive evaluation reveals their strengths and weaknesses, offering a deeper understanding of their role in abstract argumentation. These findings contribute to a more structured understanding of inconsistency measurements in abstract argumentation graphs.

In this section (Section 1) we have started with establishing the goal of this thesis and the research context by introducing and linking together the concepts of argument graphs, inconsistency measures and their characteristics. In Section 2, the Preliminaries, we examine argument graphs in greater detail, consider subsidiary definitions fundamental for this thesis, explore semantics for abstract argumentation graphs, and introduce the concept of strong equivalence. In Section 3, we inspect the inconsistency measures and their qualitative characteristics individually. This is followed by an analysis in Section 4 where the rationality postulates are investigated. We also assess the compliance of the inconsistency measures with the postulates. Finally, Section 5 concludes.

2 Preliminaries

Given the limited body of works on inconsistency measures in argument graphs in abstract argumentation so far, the foundation of this thesis is built upon works by Hunter [24] and Amgoud and Ben-Naim [1]. However, other literature such as [30] by Oikarinen and Woltran, central to four postulates proposed in this thesis, and [39] by Thimm which gives valuable insight into the characteristics of inconsistency measures, are relevant as well.

Likewise, it is important to highlight that this thesis is focused on abstract argumentation, where arguments are treated as atomic units. Atomic units refer to the fact that they cannot be further divided [10]. The emphasis is placed on the relationships between the arguments, without the need to examine their internal structure, such as premises or conclusions, as required in logic-based argumentation [24].

To fully grasp the concepts presented in this thesis, some preliminaries are essential which are given in the following sections.

2.1 Argument Graphs

In 1995, Dung introduced the concept of an abstract argumentation graph (hereafter occasionally referred to as argument graph or graph) in [17] which offers a formal reasoning framework, enabling the definition of specific semantics for argument acceptance, conflict resolution, and consistency.

Definition 2.1 A *(directed) graph* is the ordered pair $G = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is the finite set of nodes and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is the set of directed arcs [21].

Definition 2.2 An *argument graph* is a directed graph $G = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is a non-empty set of nodes representing arguments, and $\mathcal{R}$ is the set of arcs ($\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$) representing the relationships between these arguments [24].

Let Nodes(G) be the set of nodes $\mathcal{A}$ in G and let Arcs(G) be the set of arcs $\mathcal{R}$ in G. Each element $a, b, c, \ldots \in \mathcal{A}$ is called an *argument*. The ordered pair $(a, b) \in \mathcal{R}$ represents a *(direct) attack* where a attacks b. The direct attack is characterized by an arrow connecting two nodes, without any node in between. With an *indirect attack* we distinguish a second type of attack. An attack (a, d) is considered an indirect attack on d if a is at least $2n + 1$ for $n \in \mathbb{Z}^+$ arcs away from d [1]. The number of arcs must be odd because a chain of attacks or arcs consists of an alternating pattern of attacks and defenses, starting with an attack. In order for the chain to be considered an attack, it has to end with an attack. It also cannot be one arc because that describes a direct attack.

Example 2.1 Consider argument graphs B and C in Figure 2. In both of these graphs (c, b) is a direct attack, where c is the *direct attacker* of b. An indirect attack, (d, b) can be seen in argument graph C where d is the *indirect attacker*, three arcs away from b.

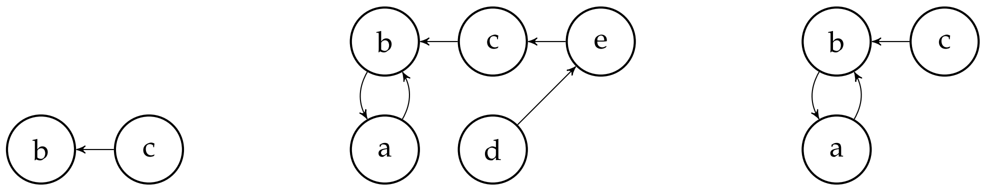


Figure 2: Argument graph B (left), C (middle), and D (right).

Definition 2.3 In a graph $G = (\mathcal{A}, \mathcal{R})$ we call an attack $(a, b) \in \mathcal{R}$ a *self-attack* or *loop* if $a = b$. We call a node $a \in \mathcal{A}$, *disconnected* or *isolated* if $\forall b \in \mathcal{A}$, $(a, b) \notin \mathcal{R}$ and $(b, a) \notin \mathcal{R}$.

Definition 2.4 Let $\mathcal{S} \subseteq \mathcal{A}$ represent a set of arguments. $\mathcal{S}$ is said to *attack* an argument $a \in \mathcal{A}$ if and only if (iff) there exists an argument $b \in \mathcal{S}$ such that (s.t.) b attacks a. Conversely, $\mathcal{S}$ *defends* an argument $a \in \mathcal{S}$ iff for every argument $b \in \mathcal{A}$, if b attacks a, it follows that $\mathcal{S}$ attacks b [24].

Definition 2.5 We consider an argument graph $G = (\mathcal{A}, \mathcal{R})$ *consistent* iff $\mathcal{R} = \emptyset$. G is defined as *inconsistent* iff $\mathcal{R} \neq \emptyset$. A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is inconsistent, denoted, $\mathcal{S} \models \perp$ iff $\exists a, b \in \mathcal{S}$ s.t. $(a, b) \in \mathcal{R}$.

2.2 Subsidiary Definitions

In this section we look at the underlying concepts used throughout this thesis.

Definition 2.6 In a graph $G = (\mathcal{A}, \mathcal{R})$, Indegree($G$, a) is the number of incoming arcs to a node $a \in \mathcal{A}$. The Outdegree(G, a) is the number of outgoing arcs from a node $a \in \mathcal{A}$ [23].

Example 2.2 Consider graph C in Figure 2. Indegree(C, b) $= 2$ and Outdegree(C, b) $= 1$.

Definition 2.7 G_1 is a *subgraph* of a graph G_2, denoted $G_1 \subseteq G_2$, iff

1. Nodes$(G_1) \subseteq$ Nodes(G_2) and
2. Arcs$(G_1) \subseteq$ Arcs$(G_2) \cap ($Nodes$(G_1) \times$ Nodes$(G_1))$ [24].

Definition 2.8 Let $N \subseteq$ Nodes(G). G' is an *induced subgraph* of G iff

1. $G' \subseteq G$,
2. Nodes(G') = N, and
3. Arcs$(G') =$ Arcs$(G) \cap (N \times N)$ [24].

Example 2.3 In Figure 2 and 3, $C' \subseteq C$. In Figure 2, D is an induced subgraph of C.

Definition 2.9 A *complete graph* is a graph where every pair of distinct nodes is connected by a unique arc and every node has a loop [16].

Definition 2.10 An *inverse graph* G^{-1} is a new graph that has the same nodes as the original graph G but the direction of all arcs is reversed [21]. The direction of loops remains unchanged.

Definition 2.11 In a graph $G = (\mathcal{A}, \mathcal{R})$ a *path* from argument a to b is a sequence $\langle a_0, \ldots, a_n \rangle$ of arguments from $\mathcal{A}$ s.t. $a_0 = a$, $a_n = b$, for any $0 \leq i < n$, $(a_i, a_{i+1}) \in \mathcal{R}$, and for all $i \neq j$, $a_i \neq a_j$. A *cycle* is a path $\langle a_0, \ldots, a_n \rangle$ s.t. $(a_n, a_0) \in \mathcal{R}$. A cycle is *elementary* iff there does not exist a cycle $\langle b_0, \ldots, b_m \rangle$ s.t. $\{b_0, \ldots, b_m\} \subset \{a_0, \ldots, a_n\}$. [1].

Definition 2.12 A graph is *acyclic* iff it does not contain any cycle [16].

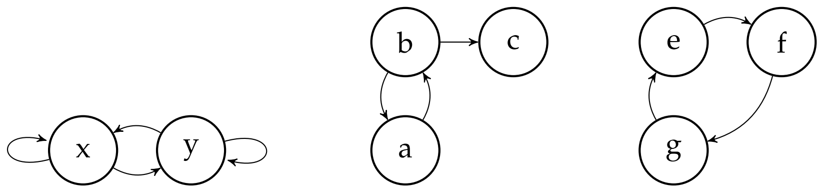


Figure 3: Argument graphs C' (left) E (middle left), F (middle right), and H (right).

Example 2.4 Figure 3 shows a complete graph E, graph $F = D^{-1}$ where D is in Figure 2, an elementary cycle H, and an acyclic graph C'.

Definition 2.13 A graph G is *isomorphic* with graph G', denoted $G \cong G'$, iff there exists a bijection φ s.t. when any two nodes a, b in G are adjacent, meaning they are connected by an arc, their corresponding nodes $\varphi(a)$, $\varphi(b)$ are also adjacent in the graph G' [23].

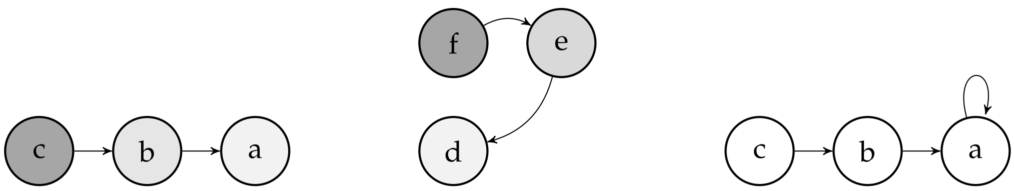


Figure 4: Argument graphs J (right), $J_{\cong}$ (middle), and J' (right).

Example 2.5 From the argument graphs in Figure 4, $J \cong J_{\cong}$, where the shaded nodes indicate the to each other corresponding and adjacent nodes.

Definition 2.14 An *undirected graph* is the ordered pair $G_{un} = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ is the finite set of nodes and $\mathcal{R} \subseteq (\mathcal{A} \times \mathcal{A})$ the set of undirected arcs [23]. A *component* of G_{un} is a maximal subset of $\mathcal{A}$ s.t. any two nodes in the subset are connected by a path in G_{un} [16].

Definition 2.15 A *component* G' of a directed graph G is a set of nodes s.t. G' is a component of the induced undirected graph G_{un}, where G_{un} is obtained from G by replacing each directed arc with an undirected arc [26].

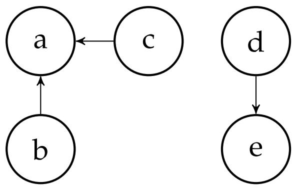


Figure 5: Argument graph K.

Example 2.6 Argument graph K in Figure 5 consists of two components, one composed of arguments $a, b,$ and c, and the other of d and e. Argument graphs J and J' in Figure 4 embody graphs with one component each.

Definition 2.16 Two graphs $G_1 = (\mathcal{A}_1, \mathcal{R}_1)$ and $G_2 = (\mathcal{A}_2, \mathcal{R}_2)$ are disjoint iff $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$ [24]. The combination of the two disjoint graphs G_1 and G_2 is denoted with $G_1 + G_2 = (\mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2)$.

Definition 2.17 For a graph $G = (\mathcal{A}, \mathcal{R})$,

- the addition of a arc $r = (a, b) \notin \mathcal{R}$ to graph G is denoted by $G + \{r\} = (\mathcal{A}, \mathcal{R} \cup \{r\}) = (\mathcal{A}, \mathcal{R} \cup \{(a, b)\})$ and
- the addition of an isolated node $a \notin \mathcal{A}$ where $\forall b \in \mathcal{A}, (a, b) \notin \mathcal{R}$ and $(b, a) \notin \mathcal{R}$, to graph G, is denoted by $G \cup \{a\} = (\mathcal{A} \cup \{a\}, \mathcal{R})$.

2.3 Semantics

Intuitively, an argument is considered acceptable if it withstands the attacks against it. The process of determining whether an argument is acceptable is called argument evaluation. There are various methods for argument evaluation, known as argumentation semantics [18]. In this thesis, extension-based semantics are significant.

Extension-based semantics, initially proposed by Dung [17], provide a framework for evaluating arguments through the usage of extensions. The aim of extension-based semantics is to determine which arguments can be collectively accepted, given their interactions. *Extensions* are sets of arguments that satisfy specific criteria within a graph. The criteria that a set of arguments must meet to be considered acceptable are specified by acceptability semantics [18]. For example, conflict-free semantics gives rise to conflict-free extensions, and similarly, admissible semantics lead to admissible extensions. Each extension therefore represents a consistent viewpoint, reflecting a set of arguments that can coexist without contradiction according to the given rules.

The extensions prevalent in this thesis are conflict-free, admissible, complete, preferred, grounded, stable, and naive extensions. The dependencies of these extensions can be seen in Figure 6, where for illustration purposes, if an extension is preferred, then it is also a complete, admissible, and conflict-free.

Definition 2.18

1. A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is *conflict-free* iff there does not exist $a, b \in \mathcal{S}$ s.t. a attacks b [17].

2. A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is *admissible* iff $\mathcal{S}$ is conflict-free and for all $a \in \mathcal{S}$, a is defended by $\mathcal{S}$ [17].

3. A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is a *complete extension* iff $\mathcal{S}$ is admissible and each argument $a \in \mathcal{A}$, which is defended by $\mathcal{S}$, belongs to $\mathcal{S}$ [17].

4. A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is a *preferred extension* iff it is a maximal (with respect to (w.r.t.) $\subseteq$) complete extension [17].

5. A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is a *grounded extension* iff it is a minimal (w.r.t. $\subseteq$) complete extension [17].

6. A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is a *stable extension* iff it is a preferred extension and $\mathcal{S}$ attacks all the arguments in $\mathcal{A} \setminus \mathcal{S}$ [17].

7. A set of arguments $S \subseteq \mathcal{A}$ is a *naive extension*, iff it is conflict-free and maximal (w.r.t. $\subseteq$) [1].

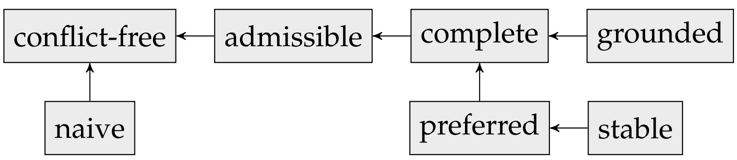


Figure 6: Implications between semantics.

The empty set is always conflict-free and admissible [17]. This is because the empty set contains no arguments, meaning it has no arguments that attack each other or themselves, allowing it to be considered defended. The grounded extension is always unique [24].

Example 2.7 Table 1 shows the extensions of argument graph B in Figure 2 under different semantics. Sets with conflict, in this case only $\{b, c\}$, are not included.

Set(B)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	×	×	×	×	×
{b}	✓	×	×	×	×	×	✓
{c}	✓	✓	✓	✓	✓	✓	✓

Table 1: Semantics of argument graph B.

Example 2.8 Table 2 shows the extensions of argument graph F in Figure 3 under different semantics. Once again, the sets with conflict, $\{a, b\}$, $\{b, c\}$, and $\{a, b, c\}$ are excluded.

Set(F)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	✓	×	✓	×	×
{a}	✓	✓	×	×	×	×	×
{b}	✓	✓	✓	✓	×	✓	✓
{c}	✓	×	×	×	×	×	×
{a, c}	✓	✓	✓	✓	×	✓	✓

Table 2: Semantics of argument graph F.

Note that in Table 1, the empty set is not a complete extension, while it is one in Table 2. This is a consequence of the presence of unattacked arguments. If there are unattacked arguments present, $\emptyset$ is not considered because there exists at least one argument that can form part of an admissible set by being conflict-free and not needing defense. Since in F every argument is attacked, $\emptyset$ is a complete extension.

2.4 Standard and Strong Equivalence

As argumentation is a dynamic process, it is significant to understand how incorporating new information impacts existing argument graphs. This is addressed by the concept of strong equivalence between argument graphs, succeeding the notion of standard equivalence [30].

Definition 2.19 Two argument graphs G and G' are *(standard) equivalent* to each other w.r.t. a semantics σ (denoted $G \equiv^{\sigma} G'$), iff $\sigma(G) = \sigma(G')$ holds [6].

Standard equivalence, which requires two argument graphs to produce the same extensions, only ensures that two graphs produce the same extensions in their current state, but it does not account for how they behave when new information is introduced. This is illustrated in Example 2.9.

Example 2.9 Consider argument graph C' in Figure 3 and L' in Figure 7. Both of these graphs have $\{b\}$ as their preferred (p), grounded (g), and stable (s) extension as visible in Tables 1 and 3, respectively, and therefore $C' \equiv^{\sigma} L'$ where $\sigma \in \{p, g, s\}$. Note that for graph C', we refer to Table 1 belonging to graph B in Figure 2, because $B \cong C'$. Thus, while the labels of the extensions are different the underlying structure and relationships are preserved. Suppose a node c is added to both graphs which attacks b, forming the graphs $J = (\mathcal{A} \cup \{c\}, \mathcal{R} \cup \{(c, b)\})$, where $\mathcal{A}, \mathcal{R}$ are from graph C' and, $J' = (\mathcal{A} \cup \{c\}, \mathcal{R} \cup \{(c, b)\})$, where $\mathcal{A}, \mathcal{R}$ are from graph L'. The graphs labeled J and J' are depicted in Figure 4 and their semantics are in Table 4 and 5, respectively. The preferred and grounded extension is $\{a, c\}$ in J but $\{c\}$ in J'. Similarly, $\{a, c\}$ is a stable extension in J but there no stable extension in J'. This shows that even though C' and L' are standard equivalent to each other w.r.t. preferred, grounded and stable semantics, they are not equivalent to each other under those semantics as new information is added.

Set(L')	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
∅	✓	✓	×	×	×	×	×
{a}	×	×	×	×	×	×	×
{b}	✓	✓	✓	✓	✓	✓	✓

Table 3: Semantics of argument graph L'.

Set(J)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
∅	✓	✓	×	×	×	×	×
{a}	✓	×	×	×	×	×	×
{b}	✓	×	×	×	×	×	✓
{c}	✓	✓	×	×	×	×	×
{a, c}	✓	✓	✓	✓	✓	✓	✓

Table 4: Semantics of argument graph J.

Set(J')	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
∅	✓	✓	×	×	×	×	×
{a}	×	×	×	×	×	×	×
{b}	✓	×	×	×	×	×	✓
{c}	✓	✓	✓	✓	✓	×	✓

Table 5: Semantics of argument graph J'.

Unlike the standard notion of equivalence, strong equivalence demands that G and G', produce the same set of extensions even when conjoined with any arbitrary graph G'' [30].

Definition 2.20 Two argument graphs G and G' are *strongly equivalent* to each other w.r.t. a semantics σ (denoted $G \equiv_s^\sigma G'$), iff for any argument graph G'', $\sigma(G \cup G'') = \sigma(G' \cup G'')$ holds [30].

To check whether two graphs are strongly equivalent, Oikarinen and Woltran [30] introduce the idea of *kernels*. In σ-kernels certain redundant attacks are removed reducing the complexity of the graph while simultaneously preserving the key structural elements of the semantics σ. In the context of this thesis, only the kernels for preferred, grounded and stable extensions are relevant; the p-kernel, g-kernel and the s-kernel, respectively. Note that the p-kernel, referred to as a-kernel in other works, serves as a uniform characterization of four different semantics, including preferred and admissible [30].

Definition 2.21 For an argumentation graph $G = (\mathcal{A}, \mathcal{R})$, Oikarinen and Woltran [30] define the

- p-kernel of G as $G^{p\mathcal{K}} = (\mathcal{A}, \mathcal{R}^{p\mathcal{K}})$, where $\mathcal{R}^{p\mathcal{K}} = \mathcal{R} \setminus \{(a,b) \mid a \neq b, (a,a) \in \mathcal{R}, \{(b,a),(b,b)\} \cap \mathcal{R} \neq \emptyset\}$,
- g-kernel of G as $G^{g\mathcal{K}} = (\mathcal{A}, \mathcal{R}^{g\mathcal{K}})$, where $\mathcal{R}^{g\mathcal{K}} = \mathcal{R} \setminus \{(a,b) \mid a \neq b, (b,b) \in \mathcal{R}, \{(a,a),(b,a)\} \cap \mathcal{R} \neq \emptyset\}$, and
- s-kernel of G as $G^{s\mathcal{K}} = (\mathcal{A}, \mathcal{R}^{s\mathcal{K}})$, where $\mathcal{R}^{s\mathcal{K}} = \mathcal{R} \setminus \{(a,b) \mid a \neq b, (a,a) \in \mathcal{R}\}$.

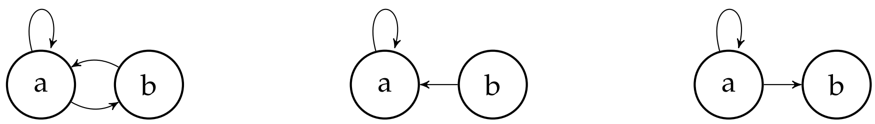


Figure 7: Argument graphs L (left), $L' = L^{p\mathcal{K}} = L^{s\mathcal{K}}$ (middle), and $L'' = L^{g\mathcal{K}}$ (right).

Example 2.10 Consider an argument graph $L = (\mathcal{A}, \mathcal{R})$ in Figure 7 where $\mathcal{A} = \{a, b\}$ and $\mathcal{R} = \{(a,a),(a,b),(b,a)\}$. By definition, the self-attack (a,a) remains in the graph for all kernels. For each kernel we iterate trough the Arcs(L) and remove the those that violate the respective semantics according to Definition 2.21.

- p-kernel: For (a,b), $a \neq b$, $(a,a) \in \mathcal{R}$, and $\{(b,a),(b,b)\} \cap \mathcal{R} \neq \emptyset$. Therefore (a,b) is removed. For (b,a), $b \neq a$, $(b,b) \notin \mathcal{R}$ s.t. (b,a) is not redundant. It follows $L^{p\mathcal{K}} = ((a,b), \{(a,a),(b,a)\}) = L'$ in Figure 7.
- g-kernel: Since for (a,b), $a \neq b$, $(b,b) \notin \mathcal{R}$ we keep (a,b) in the graph. (b,a) is removed because $b \neq a$, $(a,a) \in \mathcal{R}$, and $\{(b,b),(a,b)\} \cap \mathcal{R} \neq \emptyset$. This results $L^{g\mathcal{K}} = ((a,b), \{(a,a),(a,b)\}) = L''$ in Figure 7.
- s-kernel: For (a,b), $a \neq b$ and $(a,a) \in \mathcal{R}$. Hence, (a,b) is discarded. As for (b,a), $a \neq b$, but $(b,b) \notin \mathcal{R}$. Therefore, the attack stays. It follows that $L^{s\mathcal{K}} = ((a,b), \{(a,a),(b,a)\}) = L'$ in Figure 7.

Oikarinen and Woltran [30] establish that the σ-kernels of two arguments graphs G and G' can be utilized to determine strong equivalence between graphs G and G' w.r.t. a semantics σ. In other words, strong equivalence is reduced to kernel equivalence as visible in the following Theorem.

Theorem 2.1 [30]
$G \equiv_s^p G' \Leftrightarrow G^{p\mathcal{K}} = G'^{p\mathcal{K}}$.
$G \equiv_s^g G' \Leftrightarrow G^{g\mathcal{K}} = G'^{g\mathcal{K}}$.
$G \equiv_s^s G' \Leftrightarrow G^{s\mathcal{K}} = G'^{s\mathcal{K}}$.

Naive extensions on the other hand, do not require kernels. Two graphs G and G' are strongly equivalent under naive semantics, where $\sigma = naive$, if they have the same arguments and the same naive extensions, Extension$_n$(G) [19].

Theorem 2.2 [19]
$G \equiv_s^{naive} G' \Leftrightarrow$ Nodes(G) = Nodes(G') and Extension$_n(G)$ = Extension$_n(G)$.

3 Inconsistency Measures

In this section we consider the inconsistency measures proposed by Hunter [24] and Amgoud and Ben-Naim [1]. Each measure is comprehensively explained and rationales behind the specific calculation methods are detailed. Moreover, insight is given into the interpretation of the inconsistency measures, including the implications of high and low values, where applicable. Based on this in-depth analysis, the strengths and weaknesses of each inconsistency measure are considered, providing a perspective not yet explored.

The inconsistency measures have been grouped into Graph Extension Measures, referring to measures make use of the semantics presented in Section 2.3, and Graph Structure Measures which are based on the structure of the argument graph.

3.1 Graph Extension Measures

Graph extension measures evaluate inconsistencies using extensions and therefore are closely tied to the semantics they utilize. Two argument graphs may appear to be identically inconsistent, under one semantics but different under another.

In this section we analyze and interpret the graph extension measures, PreferredCount, NonGroundedCount, and UnstableCount proposed by Hunter [24], as well as NaiveExtensionCount and PreferredExtensionCount formulated by Amgoud and Ben-Naim [1][1].

[1]NaiveExtensionCount and PreferredExtensionCount were renamed for the sake of clarity. Amgoud and Ben-Naim [1] refer to them collectively as "Extension-based measures".

3.1.1 PreferredCount

PreferredCount (I_{pr}) counts the number of preferred extensions, $\text{Extension}_p(G)$, in a graph G and subtracts one.

$$I_{\text{pr}}(G) = |\text{Extension}_p(G)| - 1$$

By quantifying the number of preferred extensions, the measure identifies the number of maximal (w.r.t. $\subseteq$) admissible [17] sets of arguments. Each of these sets resolve the conflicts present in the graph by providing a selection of arguments that do not attack one another and are defended, and cannot be extended any further without losing their admissibility. Multiple preferred extensions may exist in a graph, representing different defended viewpoints. Although, for a definitive resolution, one preferred extensions must ultimately be selected. In other words, PreferredCount returns the number of resolutions that coexist, with one subtracted.

One is subtracted from the preferred extension quantity s.t. a single preferred extension results in an inconsistency value of 0. The measure value 0 does not indicate the absence of inconsistency in the graph, rather it signifies that there is only a single coherent way of resolving the conflict, namely by accepting the arguments in the only preferred extension.

Consequently, the higher the PreferredCount, the more diverging viewpoints or resolutions exist, and the lower the measure, the less diverging viewpoints exist.

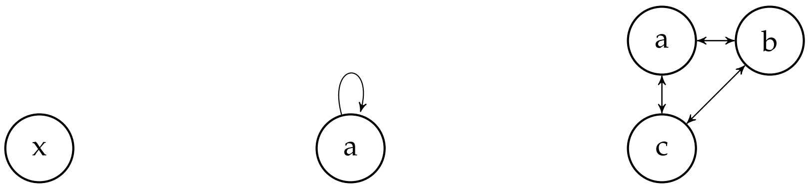


Figure 8: Argument graphs M (left), N (middle), and O (right).

Example 3.1 Observe the consistent argument graph M in Figure 8. Table 6 which summerizes the semantics for the graph, shows that M has only one preferred extension $\{x\}$. It follows $I_{\text{pr}}(M) = 1 - 1 = 0$. This is to be interpreted s.t. there is only one choice, which is to accept argument x.

Set(M)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	✗	✗	✗	✗	✗
{x}	✓	✓	✓	✓	✓	✓	✓

Table 6: Semantics for argument graph M.

Example 3.2 Consider argument graph N in Figure 8 which attacks itself. As visible in Table 7, it has only one preferred extension, namely the empty set. Hence, $I_{\text{pr}}(N) = 1 - 1 = 0$. We can interpret this result as that there is only one course of action, which is not to accept any argument.

Set(N)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
$\emptyset$	✓	✓	✓	✓	✓	✗	✓
{a}	✗	✗	✗	✗	✗	✗	✗

Table 7: Semantics for argument graph N.

Examples 3.1 and 3.2 display that despite argument graph N being inconsistent and M consistent, both argument graphs have a PreferredCount of 0. This suggests that PreferredCount, indicates the number of additional sets available for resolving conflicts alongside a selected set, rather than providing information about the overall inconsistency of the graph. Additionally, it is visible that PreferredCount does not distinguish between a non-empty preferred extension and an empty preferred extension. Example 3.3 presents a case where multiple preferred extensions exist in an argument graph, offering multiple ways of resolving conflict.

Example 3.3 Take into account argument graph O in Figure 8. According to Table 8 it has three preferred extensions $\{a\}, \{b\}$, and $\{c\}$, implying that the selection of one of these resolves the conflict in the graph. $I_{\text{pr}}(O) = 3 - 1 = 2$. We interpret this result s.t. there are two additional sets, alongside our selected one, that we can choose to resolve the conflict in the graph.

Set(O)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
$\emptyset$	✓	✓	✓	✗	✓	✗	✗
{a}	✓	✓	✓	✓	✗	✓	✓
{b}	✓	✓	✓	✓	✗	✓	✓
{c}	✓	✓	✓	✓	✗	✓	✓

Table 8: Semantics for argument graph O.

Apart from the distinct focus of PreferredCount which does not necessarily distinguish between consistent and inconsistent graphs, the measure also does not always account for differing levels of conflict because it does not consider the structure of the graph. For instance, consider graphs B in Figure 2 and E in Figure 3 with the semantics of B in Table 1 and those of E in Table 9. Both graphs have two arguments where graph B is a simple graph with one direct attack, and graph E is a complete graph with maximal conflict. Yet, $I_{\text{pr}}(B) = I_{\text{pr}}(E) = 1 - 1 = 0$.

However, this measure does not only have disadvantages. By quantifying the preferred extensions this measure already provides a fine granular assessment of the different viewpoints. This makes it especially useful as a guiding resolution strategy. Knowing that a graph has many preferred extensions can be utilized to prompt further analysis such as the application of other inconsistency measures or criteria which may function as tie-breakers to select among the alternatives [45].

Set(E)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	✓	✓	✓	×	✓
{x}	×	×	×	×	×	×	×
{y}	×	×	×	×	×	×	×

Table 9: Semantics of argument graph E.

3.1.2 NonGroundedCount

NonGroundedCount (I_{ngr}) counts the nodes in a graph G, Nodes(G), that are not in the grounded extension, Extension$_g$(G), and not attacked by a member of the grounded extension denoted by Attacked(G).

$$I_{\text{ngr}}(G) = |\text{Nodes}(G) \setminus (\text{Extension}_g(G) \cup \text{Attacked}(G))|$$

$$\text{where} \quad \text{Attacked}(G) = \{b \mid (a, b) \in \text{Arcs}(G) \text{ and } a \in \text{Extension}_g(G)\}$$

As stated in Section 2.3, a grounded extension is a unique minimal (w.r.t $\subseteq$) set of arguments that can be defended without contradiction. As the grounded extension captures the arguments in the argument graph, most of the arguments should either be attacked, excluded, or included in the grounded extension.

NonGroundedCount quantifies how many arguments in the graph are left unresolved by the grounded extension, where unresolved corresponds to the fact that the argument is neither considered acceptable nor directly attacked. The arguments that remain outside of this coverage imply potential inconsistencies. In other words, the measure reflects the spread of alternative viewpoints by examining how many extensions deviate from the minimal grounded extension. Consequently, a higher NonGroundedCount means more arguments are neither accepted nor rejected, indicating inconsistency, and a lower NonGroundedCount signifies that most arguments are effectively addressed.

Similarly to PreferredCount, a NonGroundedCount value of 0 does not necessarily suggest that the graph is consistent. Rather, it means that the every argument in the graph is either included in the grounded extension or is attacked by the grounded extension. This means all arguments are addressed in some way, either by acceptance, as is the case in Example 3.4 or by being attacked as visible in Example 3.5. Example 3.6 demonstrates a case where the NonGroundedCount is not 0.

Example 3.4 Given the semantics of argument graph M in Figure 8 in Table 6: Extension$_g$(M) = $\{x\}$, Attacked(M) = $\emptyset$, and Nodes(M) = $\{x\}$.

$$\begin{aligned} I_{\text{ngr}}(M) &= |\text{Nodes}(M) \setminus (\text{Extension}_g(M) \cup \text{Attacked}(M))| \\ &= |\{x\} \setminus (\{x\} \cup \emptyset)| \\ &= |\{x\} \setminus \{x\}| = |\emptyset| = 0 \end{aligned}$$

Example 3.5 Examine argument graph B in Figure 2 together with Table 1: Extension$_g$(B) = $\{c\}$, Attacked(B) = $\{b\}$, and Nodes(B) = $\{b, c\}$.

$$\begin{aligned} I_{\text{ngr}}(B) &= |\text{Nodes}(B) \setminus (\text{Extension}_g(B) \cup \text{Attacked}(B))| \\ &= |\{b, c\} \setminus (\{c\} \cup \{b\})| \\ &= |\{b, c\} \setminus \{b, c\}| = |\emptyset| = 0 \end{aligned}$$

Example 3.6 Consider argument graph A in Figure 1 as well as Table 10: Extension$_g$(A) = $\emptyset$, Attacked(A) = $\emptyset$, and Nodes(A) = $\{a, b\}$.

$$\begin{aligned} I_{\text{ngr}}(A) &= |\text{Nodes}(A) \setminus (\text{Extension}_g(A) \cup \text{Attacked}(A))| \\ &= |\{a, b\} \setminus (\emptyset \cup \emptyset)| \\ &= |\{a, b\} \setminus \emptyset| = |\{a, b\}| = 2 \end{aligned}$$

The NonGroundedCount of 2 signifies that there are 2 unresolved arguments with conflict in the graph, namley a and b.

Set(*A*)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
{}	✓	✓	✓	✗	✓	✗	✗
{a}	✓	✓	✓	✓	✗	✓	✓
{b}	✓	✓	✓	✓	✗	✓	✓

Table 10: Semantics for argument graph A.

When the NonGroundedCount measure quantifies the number of arguments that are left unaddressed by the grounded extension, it does so, without necessarily accounting for the component structure of the graph. In turn, if many arguments fall outside the grounded extension, this may indicate that some areas of the graph are isolated from the set of justified arguments. This is apparent in Graph $M + A$ consisting of graphs M and A in Figures 8 and 1, respectively. Based on the semantics in Table 11 we compute:

$$\begin{aligned} I_{\text{ngr}}(A + M) &= |\text{Nodes}(A + M) \setminus (\text{Extension}_g(A + M) \cup \text{Attacked}(A + M))| \\ &= |\{a, b, x\} \setminus (\{x\} \cup \emptyset)| \\ &= |\{a, b, x\} \setminus \{x\}| = |\{a, b\}| = 2. \end{aligned}$$

This result reflects the fact that nodes a and b, make up a component that is disconnected from x. While NonGroundedCount does not always offer insight into component structure it can help to identify potential gaps in the information represented within the argument graphs and pave the way for the application of other inconsistency measures.

Set(M+A)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
∅	✓	✓	×	×	×	×	×
{a}	✓	✓	×	×	×	×	×
{b}	✓	✓	×	×	×	×	×
{x}	✓	✓	✓	×	✓	×	×
{a, x}	✓	✓	✓	✓	×	✓	✓
{b, x}	✓	✓	✓	✓	×	✓	✓

Table 11: Semantics of argument graph $M + A$.

Although the NonGroundedCount is based on the grounded extension, the most cautious and well-founded extension [17], the dependency on this extension also can be problematic. For instance, in some graphs the grounded extension may be empty, making the measure unstable or less meaningful. This is because NonGroundedCount relies on arguments outside the grounded extension, and if the grounded extension is empty, then all arguments will be counted in the measure, making it meaningless. For illustration, refer to argument graph H in Figure 3 and its semantics in Table 12 which shows that the grounded extension is ∅, excluding all arguments in the graph, e, f, and g s.t.

$$\begin{aligned} I_{\text{ngr}}(H) &= |\text{Nodes}(H) \setminus (\text{Extension}_g(H) \cup \text{Attacked}(H))| \\ &= |\{e, f, g\} \setminus (\emptyset \cup \emptyset)| \\ &= |\{e, f, g\} \setminus \emptyset| = |\{e, f, g\}| = 3. \end{aligned}$$

This result merely reflects the total number of nodes in H. Likewise, any complete graph with three nodes would receive a NonGroundedCount of 3, since its grounded extension is the empty set.

Set(H)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
∅	✓	✓	✓	✓	✓	×	×
{e}	✓	×	×	×	×	×	✓
{f}	✓	×	×	×	×	×	✓
{g}	✓	×	×	×	×	×	✓

Table 12: Semantics of argument graph H.

3.1.3 UnstableCount

UnstableCount (I_{nst}) returns the minimum number of arguments that need to be removed to obtain a stable extension. As defined in Section 2.3, a set of arguments $\mathcal{S} \subseteq \mathcal{A}$ is a stable extension iff it is a preferred extension and $\mathcal{S}$ attacks all the arguments outside of itself.

To obtain UnstableCount we list the Nodes of G, and then examine different subsets $N \subseteq \text{Nodes}(G)$ as candidates for removal. For each set N, we remove those nodes from G to obtain the induced subgraph $\text{Induced}(G, N)$ which consists of the nodes $\text{Nodes}(G) \setminus N$ and all the attacks between them that were present in G. Finally, for each induced subgraph, we determine if it has at least one stable extension, denoted $\text{Extension}_s(G)$. Among all the sets N for which the induced subgraph has a stable extension we identify the one(s) with the smallest number of arguments, $|N|$. The measure $I_{\text{nst}}(G)$ is equal to this smallest number.

$$I_{\text{nst}}(G) = \min \{|N| \mid \text{Extension}_s(\text{Induced}(G, N)) \neq \emptyset \text{ s.t. } N \subseteq \text{Nodes}(G)\}$$

The idea behind this measure is to assess how far away the graph is from being stable by quantifying the minimal modification necessary to achieve this. A low UnstableCount indicates that the graph is close to being stable. In contrast, a higher value implies that a large number of arguments must be removed to obtain stability. Analogous to the previous inconsistency measures, and as visible in Example 3.7, an UnstableCount of 0 does not necessarily signify that the graph is free of inconsistency, it merely implies that there already exists a stable extension in the graph.

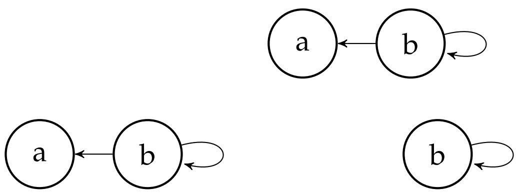


Figure 9: Argument graph P (above), $\text{Induced}(P, \emptyset)$ (left), $\text{Induced}(P, \{a\})$ (middle), and $\text{Induced}(P, \{b\})$ (right).

Example 3.7 Consider argument graph P in Figure 9. The subsets are $N = \emptyset, \{a\}, \{b\}$, and $\{a, b\}$. The induced subgraphs for these are depicted in Figure 9. For the induced subgraph $\text{Induced}(P, \emptyset)$ consider the semantics in Table 13. The semantics for the graphs $\text{Induced}(P, \{a\})$ and $\text{Induced}(P, \{b\})$ correspond to the semantics in for graphs N and M in Tables 7 and 6, respectively, as they are isomorphic to each other.

$$\begin{aligned} N = \emptyset &: \text{Extension}_s(\text{Induced}(P, \emptyset)) = \text{Extension}_s(\{a, b\}) = \emptyset \\ N = \{a\} &: \text{Extension}_s(\text{Induced}(P, \{a\})) = \text{Extension}_s(\{b\}) = \emptyset \\ N = \{b\} &: \text{Extension}_s(\text{Induced}(P, \{b\})) = \text{Extension}_s(\{a\}) = \{a\} \\ N = \{a, b\} &: \text{Extension}_s(\text{Induced}(P, \{a, b\})) = \text{Extension}_s(\emptyset) = \emptyset \end{aligned}$$

The smallest non-empty subset that satisfies the stability condition is $N = \{b\}$ with $\{a\}$. $|N| = 1$. It follows, $I_{\text{nst}}(P) = 1$. Hence, the minimal removal of 1 argument, would result in a stable extension.

Set(P)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	✓	✓	✓	×	×
{a}	✓	×	×	×	×	×	✓
{b}	×	×	×	×	×	×	×

Table 13: Semantics of argument graph P.

Example 3.8 Consider argument graph B in Figure 2. The semantics for B are evident in Table 1. The semantics for Induced($B, \{b\}$) and Induced($B, \{c\}$), are equivalent to the semantics in Table 6 since the induced graphs correspond to graph M in Figure 8.

$$N = \emptyset : \text{Extension}_s(\text{Induced}(B, \emptyset)) = \text{Extension}_s(\{b, c\}) = \{c\}$$

Since $N = \emptyset$ already results in the graph with a stable extension, it is not necessary to remove any arguments and to make the graph stable. It follows that $I_{\text{nst}}(B) = 0$.

UnstableCount is advantageous because it provides a clear measure of inconsistency by identifying how much of the graph must be altered to regain stability. Since the measure explicitly identifies the minimum number of arguments to remove, it suggests a direct way to make the graph stable.

One weakness in UnstableCount is that it only states how many arguments need to be removed but not why the graph is unstable or how strongly unstable it is. This is illustrated in Example 3.9. This example also displays how UnstableCount neglects the structure of the graph. The facts that acyclic graphs always have at least one stable extension [17] and that in an odd cycle no stable extension exists [34] show that the structural properties of a graph such as the presence or absence of cycles directly influence whether a stable extension can exist, impacting UnstableCount.

Example 3.9 Take into account graph H in Figure 3. For the graphs Induced($H, \{e\}$), Induced($H, \{f\}$), and Induced($H, \{g\}$) consider the semantics of graph M in Table 6 since each induced subgraph is isomorphic to M. Similarly, for the graphs Induced($H, \{e, f\}$), Induced($H, \{e, g\}$), and Induced($H, \{f, g\}$) consider the semantics in Table 1. Lastly, for Induced$(H, \emptyset)$ the semantics are displayed in Table 12.

$$\begin{aligned}
N = \emptyset &: \text{Extension}_s(\text{Induced}(H, \emptyset)) = \text{Extension}_s(\{e, f, g\}) = \emptyset \\
N = \{e\} &: \text{Extension}_s(\text{Induced}(H, \{e\})) = \text{Extension}_s(\{f, g\}) = \{f\} \\
N = \{f\} &: \text{Extension}_s(\text{Induced}(H, \{f\})) = \text{Extension}_s(\{e, g\}) = \{g\} \\
N = \{g\} &: \text{Extension}_s(\text{Induced}(H, \{g\})) = \text{Extension}_s(\{e, f\}) = \{e\} \\
N = \{e, f\} &: \text{Extension}_s(\text{Induced}(H, \{e, f\})) = \text{Extension}_s(\{g\}) = \{g\} \\
N = \{e, g\} &: \text{Extension}_s(\text{Induced}(H, \{e, g\})) = \text{Extension}_s(\{f\}) = \{f\} \\
N = \{f, g\} &: \text{Extension}_s(\text{Induced}(H, \{f, g\})) = \text{Extension}_s(\{e\}) = \{e\} \\
N = \{e, f, g\} &: \text{Extension}_s(\text{Induced}(H, \{e, f, g\})) = \text{Extension}_s(\emptyset) = \emptyset
\end{aligned}$$

Among all the sets N we identify, $|N| = \{f\} = \{g\} = \{e\}$ as the ones with the smallest number of arguments, with one stable extension. Therefore, $I_{\text{nst}}(H) = 1$. As apparent, there are multiple ways to remove arguments and restore stability, but no more detail is given by UnstableCount. The instability stemming from the cyclic nature of H which has an odd number of nodes is also not addressed by this measure.

3.1.4 NaiveExtensionCount

The *NaiveExtensionCount* (I_{n}) counts the number of naive extensions in a graph G, Extension$_n(G)$, adds the number of self attacks, selfAttacks(G), and subtracts one.

$$I_{\text{n}}(G) = |\text{Extension}_n(G)| + |\text{selfAttacks}(G)| - 1$$

One is subtracted to account for the case of empty attack relations s.t. if there are no attacks, the graph receives an inconsistency measure of 0 [1] as illustrated in Example 3.10. Note, that as with the previous extension-based inconsistency measures, a NaiveExtensionCount of 0 does not necessarily equate to a consistent graph. However, unlike the previous measures, self-attacks are added because their exclusion would lead to misleading inconsistency values [1]. To exemplify, consider consistent graph M and self-attacking graph N in Figure 8, which without the inclusion of self-attacks, would receive the same NaiveExtensionCount.

Example 3.10 Examine graphs N and M in Figure 8. Graph N has one naive extension, namely the empty set, as visible in Table 7 as well as one self-attack. Hence, $I_{\text{n}}(N) = 1 + 1 - 1 = 1$. Graph M also has one naive extension as apparent in Table 6, but no self-attack. It follows, $I_{\text{n}}(M) = 1 + 0 - 1 = 0$.

The idea of this measure is that if there exist multiple extensions in the graph this indicates the presence of inconsistency in the graph, where the higher the quantity of extensions in the graph, the greater the inconsistency [1]. More specifically, NaiveExtensionCount serves as a measure of the diversity of acceptable viewpoints within an argumentation graph. The higher the number of naive extensions, the more different sets of arguments can exist without conflict between them. Likewise, the lower the number of naive extensions the less sets of arguments can exist without conflict between them.

Example 3.11 Take graph B in Figure 2 as the subject of study. B has no self-attacks, and according to Table 1, it has two naive extensions. Thus, $I_{\mathrm{n}}(B) = 2 + 0 - 1 = 1$. Now consider graph E in Figure 3 and its semantics in Table 9. Because it is a complete graph, there exists no conflict free set, except for the empty set. Additionally, two self-attacks are visible s.t. $I_{\mathrm{n}}(E) = 1 + 2 - 1 = 2$. The increased conflict in E in contrast to B is reflected in the NaiveExtensionCount.

However, it needs to be acknowledged that by definition, naive extensions only require conflict-freeness and not admissibility. This means that while more naive extensions imply many conflict-free groups of arguments, they do not imply that these sets are fully defended from attacks of arguments outside. Another weakness of this measure is that two graphs with the identical total number of naive extensions and self-attacks will have an indistinguishable NaiveExtensionCount, even if their structure and the way the inconsistency manifests are very different. This can be observed in argument graph O in Figure 8, whose semantics are in Table 8 and graph E in Figure 3, for which we computed $I_{\mathrm{n}}(E) = 2$ in Example 3.11. Graph O has three naive extensions and no self-attacks s.t. $I_{\mathrm{n}}(O) = 3 + 0 - 1 = 2$. Even though graph O is a cyclic graph and E is complete graph, they receive the same NaiveExtensionCount.

Yet, this measure also has advantages. Among them is the inclusion of self-attacks as those are inherently problematic [13]. Furthermore, NaiveExtensionCount provides a differentiated view on inconsistency, because unlike other measures which capture only semantic or structural information, NaiveExtensionCount considers both, the presence of naive extensions and self-attacking arguments, capturing multiple dimensions of inconsistency.

3.1.5 PreferredExtensionCount

The measure *PreferredExtensionCount* (I_{p}) quantifies the number of preferred extensions in graph G, $\text{Extension}_p(G)$, adds number of self-attacks, selfAttacks(G) and subtracts one.

$$I_{\mathrm{p}}(G) = |\text{Extension}_p(G)| + |\text{selfAttacks}(G)| - 1$$

This measure is similar to NaiveExtensionCount, because it counts the (preferred) extensions, adds the self-attacks and subtracts one. Once again, one is subtracted to account for the case of empty attack relations [1], which is illustrated in Example 3.12. However, we emphasize once more, that a PreferredExtensionCount of 0 does not necessarily indicate that the graph is consistent. As in the NaiveExtensionCount, the self-attacks are added because they are inherently inconsistent and therefore increase the inconsistency value.

The number of preferred extensions can be interpreted as how many maximal (w.r.t. $\subseteq$) defended viewpoints exist within an argument graph. Since the number of preferred extensions refers to the number of maximal (w.r.t. $\subseteq$) admissible sets, this measure is even stricter than the NaiveExtensionCount, which focuses on maximal (w.r.t. $\subseteq$) conflict-free sets only.

It is apparent that PreferredExtensionCount also shares similarities with PreferredCount in Section 3.1.1, which quantifies the number of preferred extensions and subtracts one. A high PreferredExtensionCount indicates that the argument graph supports many defensible viewpoints without any strong consensus. In contrast, a low PreferredExtensionCount suggests that the graph has fewer defensible perspectives, indicating a higher level of agreement among the arguments.

Example 3.12 Consider argument graphs M and N in Figure 8. M has no self-attacks and one preferred extension according to Table 6. $I_{\mathrm{p}}(M) = 1 + 0 - 1 = 0$. Graph N has one preferred extension according to Table 7, and one self-attack. It follows, $I_{\mathrm{p}}(N) = 1 + 1 - 1 = 1$. N is more inconsistent than M.

Strengths of this measure include its focus on preferred extensions as these are maximal admissible sets, meaning they include as many defensible arguments as possible. By counting those, PreferredExtensionCount reflects the degree of fragmentation in justifiable viewpoints within the argument graph. Therefore this measure is useful in identifying graphs where consensus is difficult to reach. Furthermore, PreferredExtensionCount shares the advantage with NaiveExtensionCount of capturing inconsistency along the dimensions of semantics and structure through the inclusion of the self-attacks.

As a drawback, we caution that a large number of preferred extensions does not necessarily mean the graph is highly inconsistent but that there are multiple reasonable viewpoints rather than outright conflicts, which may lead to an overestimation of inconsistency. Similarly, a smaller number of preferred extensions does not necessarily mean that the graph is more consistent. Example 3.13 illustrates an underestimation of inconsistency.

Example 3.13 Observe graph H in Figure 3, which is considered inconsistent as an odd elementary cycle with no self-attacks [7]. However, according to Table 12 it has only one empty preferred extension s.t. $I_{\mathrm{p}}(V) = 1 + 0 - 1 = 0$. H receives the same measure as M from Example 3.12 despite important structural differences.

Lastly, like PreferredCount, PreferredExtensionCount is unable to differentiate between a graph that has an empty preferred extension and one that has a non-empty preferred extension. In Example 3.12 $\{x\}$ for graph M, and $\emptyset$ for graph H in Example 3.13 are not distinguished. Yet, an empty preferred extension typically indicates that no arguments can be collectively accepted, suggesting a higher level of inconsistency within the graph whereas a non-empty preferred extension means there is at least one coherent set of acceptable arguments.

3.2 Graph Structure Measures

Graph structure measures evaluate inconsistencies by considering the structure of the graph, where structure refers to the patterns of the nodes, the arcs that connect them, and the properties that emerge from them, such as cycles. Depending on what different structural features these measures assess they can be coarse or fine-grained.

In this section we examine the graph structure measures, Drastic, WeightedInSum, WeightedOutSum, CycleCount, WeightedCycleCount, WeightedComponentCount proposed by Hunter [24], as well as the Connectance Measure, the In-degree Measure and the Distance-based Measure formulated by Amgoud and Ben-Naim [1] [2].

3.2.1 Drastic

If an argument graph G has no attacks, *Drastic* (I_{dr}) returns 0, otherwise it returns 1. Thus, it analyzes the connectivity of a graph through binary classification without quantifying the amount of conflict.

$$I_{\text{dr}}(G) = \begin{cases} 0 & \text{if Arcs}(G) = \emptyset \\ 1 & \text{otherwise} \end{cases}$$

Example 3.14 For graphs B and C in Figure 2, $I_{\text{dr}}(B) = I_{\text{dr}}(C) = 1$. Graph M in Figure 8 has a drastic value of $I_{\text{dr}}(M) = 0$.

The fact that this measure assigns all graphs that have at least one attack a value of 1, and does not quantify the amount of conflict for each graph, implies that it lacks depth to a great extent. Drastic also cannot take into account the structure of the graph, such as elementary cycles and complete graphs which always receive an inconsistency value of 1 [24], nor the type of attacks including indirect attacks, self-attacks, and direct attacks.

Despite the fact that this measure has many weaknesses, it is still a useful measure and can function as an initial screening for a system. For example, when dealing with large argument graphs, this measure could be used to quickly assess whether deeper, more computationally intensive analysis is required or not. If a graph has empty attack relations, the system can skip unnecessary computational complexity, making decision-making faster and more resource-efficient.

[2] Please note that in measures like the Connectance Measure, the word "Measure" is part of their original name. In contrast, other measures, such as Drastic, do not include "Measure" in their name.

3.2.2 WeightedInSum

WeightedInSum (I_{win}) sums up the reciprocal of the indegree for every node in an arbitrary graph G.

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

The indegree must be greater or equal to one because nodes that have an indegree of zero, would be computed as $\frac{1}{0}$ which is mathematically undefined. By inverting the indegree, weight is placed onto arguments that are less attacked, making them have a higher contribution to the overall inconsistency. The rationale is that if an argument is attacked by fewer arguments, it might be more reliable or stable, which is why such attacks should contribute more to the inconsistency, in contrast to arguments that are heavily under attack and therefore more likely to be rejected.

The WeightedInSum measure captures the distribution of attacks by focusing more on the sparsely attacked arguments. If this measure is high, many arguments have a low indegree, which in turn indicates that the conflict is more wide spread. If the measure is low, few arguments have a high indegree with the implication that the conflict is more concentrated.

Example 3.15 Examine graphs R and Q in Figure 10. $I_{\text{win}}(R) = \frac{1}{3} \approx 0.33$ and $I_{\text{win}}(Q) = 1 + 1 + 1 + 1 = 4$. These values reflect the fact that the attacks in graph R are more concentrated than the ones in Q.

If the indegree for every node were to be summed up without weight, nodes with a high number of incoming attacks would contribute more to the inconsistency measure value. This could lead to a disproportionate influence of the overall inconsistency measure, by nodes that have a high number of attackers and may lead to the belief that the inconsistency is more significant than it actually is. By giving less weight to arguments that have a high number of attackers the WeightedInSum measure reduces this effect and introduces an additional layer of granularity. This strength illustrated in Example 3.16.

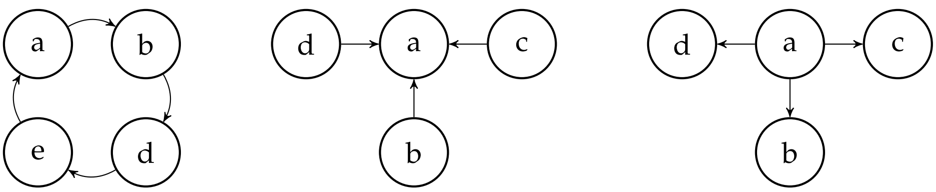


Figure 10: Argument graphs Q (left), R (middle), and R^{-1} (right).

Example 3.16 Consider graph R in Figure 10. With the exclusion of weights, summing up the indegree of every node would result in an inconsistency value of 3 since Indegree(R, a) $= 3$. In comparison, the WeightedInSum measure utilizes weights and assigns $I_{\text{win}}(R) = \frac{1}{3} \approx 0.33$. The exclusion of weights makes it seem like there exists a significant inconsistency in R, because of a's large contribution. In the WeightedInSum measure a's contribution is reduced to $\frac{1}{3}$. The weight in the WeightedInSum measures prevents the domination of the inconsistency value by a.

This described effect makes the measure especially useful in graphs that have nodes that receive many attacks. Another advantage of the weight is that since the measure reduces the impact of outliers, it makes the inconsistency value more comparable across different argument graphs. By downplaying extreme values, this approach helps in standardizing inconsistency measurements, allowing for more meaningful cross-graph comparisons. However, this also means that arguments with many attacks are given less weight, which can lead to an underestimation of the overall inconsistency in the graph. Since attacks are the source of conflict in argument graphs, reducing their influence might obscure the true level of inconsistency. As a result, while the measure offers greater generalization, it may fail to fully capture the complexity of inconsistency in cases where the conflict is concentrated in a few highly attacked arguments.

While this measure quantifies the dispersion of conflict, it has the disadvantage that it cannot necessarily distinguish between a complete graph, for which $I_{\text{win}} = 1$ always [24] and an arbitrary graph for which it is also possible to take the value 1. This is visible in graphs B in Figure 2 and graph E in Figure 3 where B is an acyclic graph and E a complete graph and $I_{\text{win}}(B) = I_{\text{win}}(E) = 1$.

3.2.3 WeightedOutSum

WeightedOutSum (I_{wout}) sums up the reciprocal of the outdegree for every node in an arbitrary graph G.

$$I_{\text{wout}}(G) = \sum_{a \in \text{Nodes}(G) \text{ s.t. Outdegree}(G,a) \geq 1} \frac{1}{\text{Outdegree}(G, a)}$$

Similar to the WeightedInSum measure the outdegree must be greater or equal to one because nodes that have an outdegree of zero, would be mathematically undefined with $\frac{1}{0}$. Once again, there is a weight placed using inversion, however, it is placed on attackers. Arguments with a higher outdegree, meaning that they attack more arguments, contribute less to the inconsistency. At the same time, arguments with a lower outdegree, meaning they attack fewer arguments, contribute more to the inconsistency. More weight is given to arguments that are less engaged in attacking other arguments. The intuition behind the weight is that arguments that attack only a few others are more focused and significant. Conversely, if an argument attacks many others, then each individual attack is less prominent in driving the inconsistency.

A high value of WeightedOutSum suggests that, on average, many arguments are involved in only a few attacks, implying that there are more focused points of conflict. Conversely, a low value of the measure tends to indicate that attackers target many different arguments. This reflects a more interconnected network where conflicts are broadly distributed.

Example 3.17 Observe graphs R and R^{-1} in Figure 10. $I_{\text{wout}}(R) = 1 + 1 + 1 = 3$ and $I_{\text{wout}}(R^{-1}) = \frac{1}{3} \approx 0.33$. The higher measure of 3 for R suggests that many different nodes are engaging in attacks, which signals the localized inconsistency present in the graph. In contrast, the lower measure of 0.33 in R^{-1} implies that different arguments are targeted indicating a more spread conflict.

WeightedOutSum is able to distinguish between a scenario where many different arguments are making only a few attacks versus a scenario where a single argument makes many attacks. This can provide insight into how the conflicts are distributed within the graph. The WeightedOutSum measure therefore is useful to identify arguments that attack many other arguments, and are disproportionately influential in driving inconsistency. WeightedOutSum also shares the advantage with WeightedInSum of downplaying extreme values but in regards to outgoing attacks.

Nevertheless, similar to WeightedInSum, the measure can oversimplify inconsistency because it does not differentiate between complete graphs which always receive a WeightedOutSum value of 1 [24] and an arbitrary graph that is not complete and also is able to receive a value of 1. To exemplify observe argument graph B in Figure 2 and complete graph E in Figure 3 for both of which $I_{\text{wout}}(B) = I_{\text{wout}}(E)$ $= 1$.

3.2.4 CycleCount

The number of cycles, Cycles(G), in an arbitrary graph G is quantified by *CycleCount* (I_{cc}).

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

A larger number of cycles often signifies greater complexity within the graph and can hint at a more a difficult resolution too as the acceptance of arguments is questionable [7] [13]. Additionally, the high number of cycles in a graph may indicate a higher connectivity which in turn could suggest that the validity of one argument might be directly tied to others in a cyclical manner. CycleCount therefore provides insight into problematic or self-reinforcing conflicts through quantifying the presence of cycles.

Based on this, a higher number of cycles points towards more inconsistency prevalent in the graph while a lower CycleCount value suggests less disagreement exists w.r.t. cycles presence. Note that CycleCount of 0 does not indicate the absence of inconsistency, merely the absence of cycles.

Example 3.18 For graph B in Figure 2 $I_{\text{cc}}(B) = 0$, as no cycle exists in the graph. For graph C in the same Figure, $I_{\text{cc}}(C) = 1$, because there exists one cycle in the graph. $I_{\text{cc}}(Q) = 1$ for graph Q in Figure 10 where Q is an elementary cycle.

The advantage of this measure is its simplicity and focus on cycles. However, through its focus, other forms of inconsistency in an acylic graph may not be detected. For instance, for acyclic argument graphs R and R^{-1} in Figure 10 $I_{\text{cc}}(R) = I_{\text{cc}}(R^{-1}) = 0$, even though there exists differently distributed inconsistency in these graphs.

CycleCount also does not differentiate between graphs that have one cycle and those that consist of an elementary cycle only. This is significant because elementary cycles may influence the evaluation of argument strength and the overall coherence of the graph differently than a more complex graph with a cycle. Furthermore, albeit the fact that CycleCount always returns a value of $I_{\text{cc}}(G) = 2^n - 1$ where $n = |\text{Nodes}(G)| > 0$ for complete graphs [24], this value is not exclusively reserved for complete graphs.

Another disadvantage of CycleCount lies in its lack of depth within its scope. As way of explanation, the measure does not distinguish between cycles of different lengths. For example for graphs Q and H in Figure 10 and 3, respectively, $I_{\text{cc}}(Q) = I_{\text{cc}}(H) = 1$, even though Q consists of only an elementary cycle of four nodes while H is an elementary cycle of odd length with three nodes. Cycle length is important because it affects argumentation semantics, determining whether conflicts can be resolved or lead to paradoxes [3] [7]. This is especially the case for odd cycles that prevent any argument from being clearly defended. Although CycleCount is a structural measure, meaning that it discards semantics, the information on cycles it returns (fails to return) can(not) further be used s.t. the observations with regard to semantics remain valid.

3.2.5 WeightedCycleCount

WeightedCycleCount (I_{wcc}) takes the reciprocal of the number of nodes involved in a cycle C in graph G, and for all cycles in G, Cycles(G), sums up the result.

$$I_{\text{wcc}}(G) = \begin{cases} 0, & \text{if Cycles}(G) = \emptyset, \\ \displaystyle\sum_{C \in \text{Cycles}(G)} \frac{1}{|C|}, & \text{otherwise.} \end{cases}$$

We extended this measure by incorporating the case Cycles$(G) = \emptyset$. The objective is that graphs with no cycles should not receive a mathematically undefined result but a WeightedCycleCount of 0. While this measure only is meaningful for graphs that have cycles, the inclusion of acyclic graphs through the addition of this case, allows for comparisons between cyclic and acyclic graphs as required in Section 4.2.

This measure accounts for the presence and size of cycles, but in a way s.t. graphs with larger cycles, meaning cycles with more nodes, contribute less to the inconsistency. By using the reciprocal, the measure gives more weight to smaller cycles. The idea behind the higher contribution of smaller cycles is that they only involve fewer arguments. As each argument is tightly connected to others in a direct way, this creates a stronger local inconsistency [1]. When a cycle involves many arguments, each argument is less directly entangled with the others in the cycle. Thus, its effect on inconsistency is considered weaker.

A high WeightedCycleCount therefore means that the graph has small cycles, while a low value of this measure means that there are typically fewer larger cycles.

Example 3.19 Refer to graph A in Figure 1, that consists of an elementary cycle with two nodes s.t. $I_{\text{wcc}}(A) = \frac{1}{2} = 0.50$. For graph Q, an elementary cycle with four nodes in Figure 10, $I_{\text{wcc}}(Q) = \frac{1}{4} = 0.25$. It is visible that since graph A involves less nodes in the cycle, has a higher measure than Q, reflecting the tighter attack loop that exists in A.

Hunter [24] generalizes the WeightedCycleCount for a complete graph G where $\text{Nodes}(G) = n$ and $n > 0$. For each non-empty subset of nodes, a cycle is constituted. For each k where $1 \leq k \leq n$, there are $\frac{n!}{k!(n-k)!}$ subsets of cardinality k, and each of these contributes $\frac{1}{k}$ to the sum. Hence, the WeightedCycleCount for a complete graph is

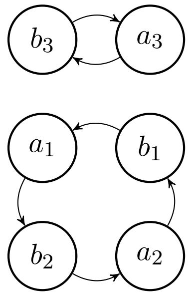
$$I_{\text{wcc}}(G) = \sum_{k=1}^{n} \left[\frac{1}{k} \times \frac{n!}{k!(n-k)!} \right].$$

Example 3.20 For a complete graph E in Figure 3 for which $n = 2$,

$$\begin{aligned} I_{\text{wcc}}(E) &= \left[\frac{1}{1} \times \frac{2!}{1!(2-1)!} \right] + \left[\frac{1}{2} \times \frac{2!}{2!(2-2)!} \right] \\ &= \left[1 \times \frac{2}{1 \times 1} \right] + \left[\frac{1}{2} \times \frac{2}{2 \times 1} \right] \\ &= [1 \times 2] + \left[\frac{1}{2} \times 1 \right] \\ &= \frac{5}{2} = 2.50. \end{aligned}$$

One strength exhibited by WeightedCycleCount is the fact that it is sensitive to the size of cycles. This aligns with the idea that smaller cycles are considered more conflicting [1]. The incorporation of cycle size also reduces the risk of overestimating inconsistency in cases where cycles occur due to the debate's interconnected structure rather than logical conflict. A structural cycle occurs when the attack-relations are cyclic simply because the discussion is complex and many arguments happen to relate to one another. Such a cycle does not necessarily mean that the positions cannot be reconciled as it just reflects the connected nature of the discussion. This is especially the case in larger cycles.

To illustrate this point, consider graph S in Figure 11 and by way of example the arguments assigned to it. The cycle consisting of nodes a_1, b_1, a_2, and b_2 is structural because each argument shifts the debate's framing from prioritizing wind energy, to prioritizing stability, to investing in nuclear energy, without directly refuting factual claims. In contrast the cycle consisting of the nodes a_3 and b_3 is due to logical conflict because the arguments contradict each other's empirical claims on property value decline. In this example it is evident that the larger cycle in the graph is due to the interconnected structure of the debate. The CycleCount measure from the previous section gives a misleading impression of the inconsistency with $I_{\text{cc}}(S) = 2$. WeightedCycleCound adresses this weakness by assigning $I_{\text{wcc}}(S) = \frac{1}{2} + \frac{1}{4} = \frac{3}{4} = 0.75$.



a_1: "Governments should prioritize wind energy investments to phase out fossil fuels, as they are a scalable and renewable technology."
a_2: "Grid modernization should be the focus to enable renewable integration, not delaying wind projects."
a_3: "Wind projects create long-term jobs, tax revenue, and infrastructure, with studies showing no decline in property values."
b_1: "Policymakers must prioritize grid stability over rapid renewable adoption since irregular sources like wind risk blackouts."
b_2: "Investments should target proven industries like nuclear first, not speculative grid upgrades for wind."
b_3: "Data confirms that wind turbines reduce property values by 10–20% in nearby areas, outweighing short-term economic gains."

Figure 11: Argument graph S.

While WeightedCycleCount, takes into account the length of cycles, note that it does not differentiate between odd and even cycles, whose importance we have discussed in Section 3.2.4 under CycleCount. Among its disadvantages also is its exclusive focus on cycles. All argument graphs that do not have cycles get an inconsistency measure of 0. It follows automatically then that this measure does not take into account the structure of the graph. For instance, consider graph C in Figure 2 where, $I_{\text{wcc}}(C) = \frac{1}{2} = 0.50$, because there exist two nodes involved in one cycle in the graph. However, WeightedCycleCount disregards the inconsistency from attacks $(d, e), (e, c),$ and (c, b) as well as the indirect attack (d, b), and focuses exclusively on the cycle between nodes a and b. This can be misleading, since graph A with $I_{\text{wcc}}(A) = 0.50$ as shown in Example 3.19, has the same inconsistency measure of C, despite its different structure as an elementary cycle.

3.2.6 WeightedComponentCount

WeightedComponentCount (I_{wco}) takes the sum of the cardinality of each component X in the graph G from all components in the graph, Components(G), subtracts one, and squares it.

$$I_{\text{wco}}(G) = \sum_{X \in \text{Components}(G)} (|X| - 1)^2$$

The subtraction of one from the quantity of nodes in the component is to ensure that the component is connected, because for n nodes where $n \geq 1$, we need at least $n - 1$ arcs s.t. there exists a minimally connected component [16]. Additionally, the subtraction ensures that a single disconnected node does not contribute to the inconsistency. The result of this cardinality is squared for each component to amplify the impact of larger components, since they represent inconsistency in larger clusters that is more difficult to resolve.

Overall, the WeightedComponentCount aims to measure how the nodes are distributed among the connected components in a graph while emphasizing the impact of larger components, which are more significant for inconsistency. By giving more weight to larger connected components, it also assesses whether the graph is mostly a single connected unit or split into many smaller, disconnected parts.

Example 3.21 Consider graphs B and C in Figure 2, each consisting of one component. $I_{\text{wco}}(B) = (2-1)^2 = 1$ and $I_{\text{wco}}(C) = (5-1)^2 = 16$. The WeightedComponentCount of 1 for B suggests that the graph consists of a smaller component. The value of 16 from graph C indicates the existence of a larger connected component. The larger value shows that the graph has more structure and greater connectivity compared to graph B.

Example 3.22 Observe graph K in Figure 5. It consists of two components, hence, $I_{\text{wco}}(K) = (3-1)^2 + (2-1)^2 = 4 + 1 = 5$. In comparison to graph C mentioned in Example 3.21, it has relatively simple structure. Although it consists of one more component than B, its components are not complex, which is why it does not contribute as much to the inconsistency.

Unlike other measures, this measure considers components of the graphs, where as the number of larger components increases, the graph becomes more inconsistent. Another strength of this measure is that it distinguishes between components s.t. a graph with a larger component and several smaller components will be assigned a higher inconsistency value than a graph consisting of smaller components only. As an example, observe argument graphs T and T' in Figures 12 and 13 which both have eight nodes. Graph T consists of five smaller components with $I_{\text{wco}}(T) = (3 \times (2-1)^2) + (2 \times (1 - 1^2)) = 3$ while graph T' consists of one larger component and four smaller components s.t. $I_{\text{wco}}(T') = (4-1)^2 + (4 \times (1-1)^2) = 9$. As visible the quadratic growth connected to component size, provides a smooth, non-linear penalty for component size.

While the subtraction of one from the number of nodes in each component ensures that an isolated node does not contribute to inconsistency, this measure also automatically discards isolated nodes that have a self-attack and therefore would contribute to inconsistency. However, we also accentuate that whether an isolated node with a self-attack should contribute to an inconsistency value or not is debatable [5] [32] [13].

Another weakness of this measure is the fact that it only depends on the size of each component. It does not capture other factors such as the structure of the graph and therefore does not discern between elementary cyles and complete graphs that have the same number of nodes. This disregard of structural information can lead to misleading results when evaluating consistency. To illustrate, consider graph A, an elementary cycle in Figure 1 and graph E, a complete graph in Figure 3, where for complete graphs it always holds that $I_{\text{wco}}(G) = (n-1)^2$ where $n = |\text{Nodes}(G)| > 0$ [24]. Both graphs consist of one component with two nodes and therefore receive an equal inconsistency value of $I_{\text{wco}}(A) = I_{\text{wco}}(E) = (2-1)^2 = 1$, despite their structural differences.

3.2.7 Connectance Measure

The *Connectance Measure* (I_{con}), quantifies the number of attacks in a graph $G = (\mathcal{A}, \mathcal{R})$, by counting the arcs.

$$I_{\text{con}}(G) = |\mathcal{R}|$$

The idea behind this measure is that inconsistencies stem from attacks [1]. The higher the Connectance Measure the more attacks exist in the graph and the more conflict is present.

Example 3.23 Take into account graphs B and C in Figure 2. $I_{\text{con}}(B) = 1$ because there exists only one attack going from node c to b and $I_{\text{con}}(C) = 5$ based on the five direct attacks. It is visible that the indirect attack (d, b) of graph C is ignored, and that the measure only counts direct attacks.

As inconsistencies stem from attacks, the Connectance Measure is a very natural and straightforward measure. Additionally it is easy to interpret and able to distinguish a consistent graph with no attacks from an inconsistent graph by definition with a Connectance Measure value of 0. However, this measure has more weaknesses than it has strengths.

Firstly, it does not take into account indirect attacks as shown in Example 3.23, nor does it distinguish between direct attacks and self-attacks. For illustration purposes, observe graph N in Figure 8, which has the same Connectance Measure value of 1 as B from Example 3.23, even though B has a direct attack and N has one self-attack.

This is misleading because according to Beuselinck et al. [12] there is a significant difference between an argument attacking itself and an argument that is attacked by another argument. A self-attack indicates an internal contradiction, whereas an external attack reflects a typical adversarial relationship between different arguments. This implies that the arc-counting measure oversimplifies the situation, missing the structural differences that have important implications for the evaluation of arguments in the graph.

The measure further lacks depth in that it does not consider the structure of the graph as it does not distinguish between elementary cycles, acyclic and complete graphs. One such situation can be seen in graph E in Figure 3, and Q in Figure 10, where E is a complete graph and Q is an elementary cycle. Since for complete graphs, it stands that $I_{\text{con}}(G) = n^2$ where $n = |\text{Nodes}(G)| > 0$ [24], $I_{\text{con}}(E) = I_{\text{con}}(Q) = 4$. Both types of graphs have the same number of arcs, but the complete graph represents a much more complex and contentious scenario than the elementary cycle. The measure treats them as equivalent, missing crucial information about how the arguments are interacting.

3.2.8 In-degree Measure

The *In-degree Measure* (I_{ind}) counts the number of arguments $a \in \mathcal{A}$ attacked in a graph G = ($\mathcal{A}$, $\mathcal{R}$).

$$I_{\text{ind}}(G) = |\{a \in \mathcal{A} \mid \exists(x, a) \in \mathcal{R}\}|$$

The quantification of attacked arguments, reflects the level of inconsistency in the graph because attacked arguments represent points of conflict. If many arguments are attacked, this suggests that a significant portion of the graph is involved in disagreements. Thus, a higher value of this measure indicates that the graph is the more inconsistent while a lower value implies a more consistent graph [1].

Example 3.24 Consider graphs B and C in Figure 2. $I_{\text{ind}}(B) = 1$ as only argument b is attacked and $I_{\text{ind}}(C) = 4$ since four arguments are attacked. The indirect attack (d, b) in C is not taken into account.

As attacked arguments represent points of disagreement, the measure directly highlights the degree of conflict. An advantage of this straightforward approach is that it avoids being misleading, since an attacked argument clearly signals a challenge to its validity. Additionally the In-degree Measure is able to differentiate between consistent and inconsistent graphs because a consistent graph always receives an In-degree Measure of 0 while an inconsistent graph cannot receive such value. While this measure can serve as a baseline indicator in that it can quickly flag whether a graph has potential issues, it also has disadvantages.

As visible in Example 3.24, the In-degree Measure does not consider indirect attacks. It also does not differentiate between different types of attacks such as direct attacks and self-attacks. A case like this can be seen with graphs B in Figure 2 and N in Figure 8 where $I_{\text{ind}}(B) = I_{\text{ind}}(N) = 1$, even though graph B has a direct attack and N has a self-attack. We have underscored the significance of the distinction between direct and self-attacks under the Connectance Measure in Section 3.2.7.

Furthermore, the simplicity of this measure comes with the cost of disregarding the structure of the graph. As such, the In-degree Measure is unable to distinguish between graphs with structural differences. In fact, any two graphs that have the same number of attacked arguments receive the same inconsistency measure. Example 3.25 portrays such a situation where an elemenatary cycle and a complete graph receive an identical In-degree measure as for both it holds that $I_{\text{ind}}(G) = |\text{Nodes}(G)|$ [1].

Example 3.25 Consider graph E in Figure 3 and graph A in Figure 1 where E is a complete graph and A is an elementary cycle. In both of these graphs two nodes are attacked. Thus, $I_{\text{ind}}(E) = I_{\text{ind}}(A) = 2$.

3.2.9 Distance-based Measure

The *Distance-based Measure* (I_{db}) is different to all previously mentioned measures in that it is the only measure that takes into account indirect attacks. It is calculated in several steps.

In a graph $G = (\mathcal{A}, \mathcal{R})$, for any pair of nodes $a, b \in \mathcal{A}$, we determine whether there exists a path between them. The length of the shortest path, indicating the number of arcs from node a to b, is termed the *distance* and is represented as $d(a, b)$. Each distance is computed differently depending on factors such as whether a node is reachable or not and other relevant criteria, summarized below [1].

1. If node b is reachable from node a then $d(a, b)$, the smallest number of arcs from node a to b, is the shortest distance between them.
2. If node b is not reachable from a, suggesting that no path exists between them, then $d(a, b) = k$ where $k = |\mathcal{A}| + 1$, since the longest path in an argument graph is $|\mathcal{A}| - 1$ and the length of the longest cycle is $|\mathcal{A}|$.
3. If $a = b$, meaning that we are considering the path from a node to itself, then $d(a, b)$ is the length of the shortest elementary cycle in which a is involved in. If a is not be involved in any cycle, $d(a, b) = k$.

The global distance, $GD(G)$, is defined as the sum of all the distances between all node pairs in the graph.

$$GD(G) = \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{A}} d(a, b)$$

The global distance provides a value of how interconnected the graph is through the summation of all the shortest path between all pairs of nodes, while also taking into account unreachable nodes and self-attacks. The lower the global distance, the more connected is the graph while the higher the global distance, the more the arguments are isolated.

Afterwards we calculate the lower (min) and upper (max) bounds for the global distance, which can be interpreted as the range within which the global distance of an argument graph falls.

$$\min \leq GD(G) \leq \max \quad \text{where} \quad \max = n^2 \times (n+1), \quad \min = n^2, \quad n = |\mathcal{A}|$$

The global distance is maximal when G is consistent, meaning there are no arcs between the nodes [1]. This is because there are n^2 ordered pairs of nodes, where each node is unreachable for every other node including itself. All of these pairs have a distance of $k = n + 1$ s.t. $\max = n^2 \times (n + 1)$. Similarly, the global distance is minimal in the case of a complete graph [1] where there exist n^2 arcs of length 1.

Finally, I_{db} can be calculated using the global distance and the upper and lower bounds. The distance-based measure evaluates how close the global distance of an argument graph is to the upper bound. The closer it is to the upper bound the less inconsistent is the graph, and the closer it is to the lower bound the more disagreement exists in the graph [1]. The underlying idea is that the closer the nodes are, meaning the smaller the distance between them, the more they attack each other. Conversely, the further apart they are, the less they attack each other, resulting in less disagreement.

$$I_{\text{db}}(G) = \frac{\max - GD(G)}{\max - \min}$$

The measure I_{db} ranges from 0 to 1. The closer I_{db} is to 0 the less inconsistent is the graph where 0 describes a consistent graph. And the closer I_{db} is to 1 the more inconsistent is the graph where 1 represents a complete graph [1]. Amgoud and Ben-Naim further show that for an elementary cycle G it holds that $I_{\text{db}}(G) \in (\frac{1}{2}, 1]$ and that for an acyclic graph G', $I_{\text{db}}(G') \in [0, \frac{1}{2})$. The fact that the Distance-based Measure ranges from 0 to 1 and always assigns consistent graphs a measure of 0 and maximally inconsistent graphs a value of 1, and also reserves ranges of certain values of acyclic graphs and elementary cycles, is beneficial for comparisons across different graphs.

Example 3.26 Analyze argument graph D in Figure 2. We consider all pairs of nodes to calculate the global distance $GD(D)$. Starting with node b we see that c is not reachable from b, hence, $d(b, c) = k = |A| + 1 = 3 + 1 = 4$. We see that a is reachable from b with one arc and therefore $d(b, a) = 1$. Going from b to b, b is involved in an elementary cycle with a which takes two arcs thus, $d(b, b) = 2$. Continuing with c, it is visible that b is reachable from c with one arc, s.t. $d(c, b) = 1$. Node a also is reachable from c with two arcs with one going over b, thus, $d(c, a)$

$= 2$. From c to c, it is evident that c is not involved in an elementary cycle, therefore $d(c, c) = k = 4$. Observe the last node a. Node c is not reachable from a, hence, $d(a, c) = k = 4$. Node b is reachable from a with one arc, thus $d(a, b) = 1$. Finally going from a to a, since a is involved in an elementary cycle of two arcs, $d(a, a) = 2$.

$$
\begin{aligned}
GD(D) &= d(b,c) + d(b,a) + d(b,b) + d(c,b) + d(c,a) + d(c,c) \\
&\quad + d(a,c) + d(a,b) + d(a,a) \\
&= 4 + 1 + 2 + 1 + 2 + 4 + 4 + 1 + 2 = 21
\end{aligned}
$$

$$
\begin{aligned}
\max &= n^2 \times (n+1) = 3^2 \times (3+1) = 9 \times 4 = 36 \\
\min &= n^2 = 3^2 = 9 \\
\min \leq GD(D) \leq \max \quad &= \quad 9 \leq 21 \leq 36 \\
I_{\text{db}}(D) = \frac{\max - GD(D)}{\max - \min} &= \frac{36-21}{36-9} = \frac{5}{9} \approx 0.56
\end{aligned}
$$

The value of 0.56 implies that disagreement is present in the graph to an extent. Since the value is closer to one than it is to zero, graph D is slightly more inconsistent than consistent.

Example 3.27 Take into account graph N in Figure 8, that consists of only one node a with a self-attack.

$$
\begin{aligned}
GD(N) &= d(a,a) = 1 \\
\max &= n^2 \times (n+1) = 1^2 \times (1+1) = 2 \\
\min &= n^2 = 1^2 = 1 \\
\min \leq GD(N) \leq \max \quad &= \quad 1 \leq 1 \leq 2 \\
I_{\text{db}}(N) = \frac{\max - GD(N)}{\max - \min} &= \frac{2-1}{2-1} = \frac{1}{1} = 1
\end{aligned}
$$

The measure of 1 suggests that the compete graph N is maximally inconsistent.

A strength of the Distance-based Measure is the fact that it considers direct and indirect attacks. Additionally, the measure considers all shortest paths in a graph, which allows for a more detailed assessment of graphs based on their level of inconsistency. For instance, two graphs could have the same number of direct attacks, but one might have more indirect attack chains, making it more conflicting. These two graphs would receive different measures from the Distance-based Measure while other more simple measures such as the Connectance Measure might treat them as equally conflicting.

While the Distance-based Measure gives insight into the connectivity of the graph, the inclusion of distances over all pairs of nodes might obscure the fact that not all indirect attacks contribute equally as hinted at by [27] and [22]. Typically, longer indirect attacks might have diminishing effects, yet the measure does not take this into account. Instead, it weighs all indirect attacks uniformly, potentially overestimating their impact.

An advantage of this measure is that it treats smaller cycles as highly conflicting. Amgoud and Ben-Naim [1] consider conflict in a larger cycle weaker than in a smaller cycle because in larger cycles the conflict is distributed or diluted across many arguments, while in smaller cycles the conflict is concentrated among few arguments. Example 3.28 showcases this with two differently sized elementary cycles.

Example 3.28 Observe two elementary cycles A in Figure 1 and Q in Figure 10.

$$\begin{aligned} GD(A) &= d(a,a) + d(a,b) + d(b,b) + d(b,a) \\ &= 1 + 2 + 2 + 1 = 6 \end{aligned}$$

$$\begin{aligned} \max &= n^2 \times (n+1) = 2^2 \times (2+1) = 4 \times 3 = 12 \\ \min &= n^2 = 2^2 = 4 \end{aligned}$$

$$\min \leq GD(A) \leq \max \quad = \quad 4 \leq 8 \leq 12$$

$$I_{\text{db}}(A) = \frac{\max - GD(A)}{\max - \min} = \frac{12-6}{12-4} = \frac{3}{4} = 0.75$$

$$\begin{aligned} GD(Q) = \; & d(a,a) + d(a,b) + d(a,d) + d(a,e) + d(b,a) + d(b,b) + d(b,d) + d(b,e) \\ & + d(d,a) + d(d,b) + d(d,d) + d(d,e) + d(e,a) + d(e,b) + d(e,d) + d(e,e) \\ = \; & 4 \times (1 + 2 + 3 + 4) = 40 \end{aligned}$$

$$\begin{aligned} \max &= n^2 \times (n+1) = 4^2 \times (4+1) = 16 \times 5 = 80 \\ \min &= n^2 = 4^2 = 16 \end{aligned}$$

$$\min \leq GD(Q) \leq \max \quad = \quad 16 \leq 40 \leq 80$$

$$I_{\text{db}}(Q) = \frac{\max - GD(Q)}{\max - \min} = \frac{80-40}{80-16} = \frac{5}{8} \approx 0.63$$

As visible, the larger cycle Q receives a smaller inconsistency value than the smaller cycle A. And for both elementary cycles it holds that $I_{\text{db}}(A), I_{\text{db}}(Q) \in (\frac{1}{2}, 1]$.

4 Analysis

In the following, we examine 18 rationality postulates for inconsistency measures in abstract argumentation graphs. These rationality postulates, also referred to as principles or postulates, have been proposed to give general guidelines on how inconsistency measures should behave in certain scenarios [38]. They provide a structured framework for assessing the measures, ensuring that they are meaningful, reliable, and applicable across different contexts.

To enhance readability in Section 4.1 and amid the numerous proofs in Section 4.2, in which we investigate which of the principles are satisfied or violated by the inconsistency measures, we have organized and labeled the postulates into the following sets visible in Table 14. The sets are explained in the table. The newly formulated principles are Penalty, Free-Node Dilution, p-Exchange, g-Exchange, s-Exchange, and $naive$-Exchange.

Postulate Set	Included Postulates	Explanation
Basic	• Consistency • Normalization • Contradiction	Ensure fundamental coherence and standardization of the measures.
Expansion	• Monotonicity • Freeness • Penalty • Free-Node Dilution	Deal with the effects of adding new elements such as arguments or arcs to graphs to ensure that measures respond appropriately to structural expansions.
Structural	• Inversion • Isomorphic Invariance	Ensure that evaluations of the measures are determined by the structural properties of the graph.
Strong Equivalence	• p-Exchange • g-Exchange • s-Exchange • $naive$-Exchange	Focus on the preservation of strong equivalence under the corresponding semantics.
Additivity	• Disjoint Additivity • Super Additivity	Ensure that evaluations account for the combination and separation of argument graphs.
Cyclicality	• Reinforcement • Cycle Precedence • Size Sensitivity	Concentrate on acyclic graphs and/or elementary cycles.

Table 14: Overview of rationality postulates and their sets.

4.1 Rationality Postulates

In this section, we explain each principle, discuss its relevance, and explore its underlying idea. Where applicable, we also interpret the implications of a principle being satisfied or violated.

4.1.1 Basic Postulates

The first principle, Consistency [24] or Agreement [1], originally proposed by Hunter and Konieczny [25], states that any graph that contains no attacks, receives an inconsistency measure of 0. This ensures that consistent graphs, receive a minimal inconsistency value of 0 and that they have a strictly positive inconsistency value [39]. It is therefore considered a minimal requirement for any reasonable inconsistency measure and is satisfied by all known concrete approaches [40].

Principle 1. *Consistency/Agreement*: If Arcs(G) $= \emptyset$, then $I(G) = 0$.

The second principle, Normalization [40], states that the inconsistent measure should be between 0 and 1, or take these values. This postulate is useful because it ensures that the values of the inconsistency measures are easy to interpret and to compare to each other, even when the argument graphs have different sizes [39]. However, Thimm also declares that this principle is usually regarded as optional based on the fact that many measures tend to assess inconsistency absolutely and not relatively [39]. This distinction, originally made in [20] states that absolute measures disregard the size of the argument graph, thus answering the question "How much inconsistency is in the argument graph?" while relative measures quantify the inconsistency in the graph w.r.t. the graph size, answering the question "How inconsistent is the argument graph?" [11].

Principle 2. *Normalization*: $0 \leq I(G) \leq 1$ where 0 represents complete consistency and 1 represents complete inconsistency.

The third principle, Contradiction [39], states that an arbitrary graph G is maximally inconsistent, with an inconsistency measure of 1, iff every possible non-empty subset $\mathcal{S}$ of its arguments $\mathcal{A}$ is also inconsistent, where the entailment of inconsistency, denoted with $\models\perp$, holds iff $\exists a, b \in \mathcal{S}$ s.t. $(a, b) \in \mathcal{R}$. The inconsistency of every subset of arguments refers to the idea that every argument is attacked by every other argument including itself, in other words, $\mathcal{A}$ is totally inconsistent, essentially describing a complete graph. Since Contradiction requires every non-empty subset of the arguments to be inconsistent, it captures an extreme case where no meaningful argumentation is possible and sets a boundary for when an argumentation graph is completely unreliable, and needs to be modified or repaired. This postulate is an extension of the Normalization principle, and only is reasonable if Normalization applies [39]. It extends Normalization by giving a precise meaning to when an argumentation graph reaches maximal inconsistency, transforming the abstract idea of normalization, which establishes a bounded scale for the measures, into a structural rule based on subsets of arguments.

Principle 3. *Contradiction*: $I(G) = 1 \Leftrightarrow \forall \mathcal{S} \subseteq \mathcal{A},\ \mathcal{S} \neq \emptyset$, and $\mathcal{S} \models \perp$.

4.1.2 Expansion Postulates

The fourth principle, Monotonicity [24] states that the addition of arguments (and attacks) to an argument graph, cannot decrease the inconsistency value. This postulate aligns with the natural intuition that adding more conflicting information could only increase or preserve the level of inconsistency, not reduce it. Measures that satisfy Monotonicity are considered to behave in a rational and predictable way. Measures that do not satisfy this principle have counter-intuitive behavior, and could give rise to misleading or unexpected results.

Principle 4. *Monotonicity*: If $G \subseteq G'$, then $I(G) \leq I(G')$.

The fifth principle, Freeness [24] or Dummy, [1] states that adding arguments that do not interact with the other arguments, do not modify the inconsistency value. The idea behind this principle is that disconnected nodes do not participate in inconsistencies and should not contribute to having a certain inconsistency value [35]. Measures should only quantify inconsistency that arises from actual conflicts in the argumentation structure [39]. Measures that satisfy Freeness therefore are focused on the actual conflicting relationships between arguments, and not influenced by irrelevant factors. Additionally, if an inconsistency measure satisfies this principle, it can be considered intuitive because it acknowledges that the inconsistency of a graph is as large as before when a new isolated argument is introduced. Measures that do not satisfy this principle can be considered counter-intuitive.

Principle 5. *Freeness/Dummy*: If Nodes(G) = Nodes(G') $\setminus$ $\{\mathcal{A}\}$ and Arcs(G) = Arcs(G'), then $I(G) = I(G')$.

The sixth principle, Penalty, inspired by [35], is the counterpart of Freeness [36] and states that adding a new arc, r to an arbitrary argument graph G must increase the inconsistency value. The addition of the arc r to graph G is denoted with $+$ and formally defined in Defition 2.17 in Section 2.2. This postulate aligns with the understanding that introducing more arcs, representing attacks, makes the argument graph more inconsistent. While the violation of this postulate can hint at the misleading nature of an inconsistency measure, it also may be that the measure's focus is on semantics where the acceptability of an argument does not change with the addition of an attack or that its focus is on structural properties that do not involve attacks. This implies that the violation of this postulate is only problematic if the inconsistency measure claims to capture inconsistency while basing itself on attacks of arguments, but does not adhere to Penalty. In turn, the satisfaction of this measure implies that the measure concentrates on attack-based conflict, and is predictable. Penalty thus can serve as a guideline for designing well-behaved inconsistency measures that focus specifically on direct attacks between arguments.

Principle 6. *Penalty*: $I(G) < I(G + \{r\})$ where r is an arc.

The seventh principle, Free-Node Dilution, inspired by [28], states that adding a new disconnected node a, meaning a node which is not attacked and does not attack other arguments or itself, to an arbitrary graph G should not increase the inconsistency value, where the addition of the node a to a graph G is denoted utilizing $\cup$ and formally defined in Definition 2.17 in Section 2.2. Applying the idea of [39], it can be claimed that this principle serves as a weaker version of Freeness. Free-Node Dilution is relevant because it takes into account the dilution of inconsistency. Unlike Freeness, it address the case of a possible decrease in the inconsistency value after the addition of a disconnected node, given that the measure quantifies relative inconsistency [39].

Principle 7. *Free-Node Dilution*: $I(G) \geq I(G \cup \{a\})$ where a is a disconnected node.

4.1.3 Structural Postulates

The eighth principle, Inversion [24], states that if G^{-1} is a graph G with its attacks inverted, where self-attacks remain the same, they both have the same inconsistency value. With this postulate it is ensured that the inconsistency is a structural property of the argument graph, and not only relating to the direction of attacks. Measures that satisfy this principle therefore treat the inconsistency as a property of the overall structure of the graph and not only tied to the attack direction, in constrast to measures that do not satisfy this postulate. However, a violation of this principle might be justified in contexts where the direction of attacks is relevant.

Principle 8. *Inversion*: If G^{-1} is the inversion of G, then $I(G) = I(G^{-1})$.

The ninth principle, Isomorphic Invariance [24] or Anonimity [1], states that if two graphs G and G' are isomorphic, they have the same inconsistency value. This postulate is useful in that it ensures that the measures are not affected by the label or representation of the graph, but the underlying structure of the graph. Hence, measures that satisfy this principle suggest that they consider the structure of the argumentation graphs and not on the specific labels or names of the arguments. Inconsistency measures that violate this postulate do not follow abstraction [41].

Principle 9. *Isomorphic Invariance/Anonymity*: If $G \cong G'$, then $I(G) = I(G')$.

4.1.4 Strong Equivalence Postulates

Principles ten to thirteen, inspired by [9], [30], and [19], state that if two argument graphs G and G' are strongly equivalent to each other w.r.t. semantics σ where, σ is preferred (p), grounded (g), stable (s), or naive ($naive$), their inconsistency remains the same. Recall that strong equivalence refers to the fact that the extensions of argument graphs G and G' stay constant even if for each argument graph an arbitrary graph G'' is added. The idea behind this principle is that replacing a consistent part of the argument graph G with an equivalent subgraph G' or vice versa should preserve equality between the inconsistency measures, even in the presence of new information. This in turn is based on the idea that exchanging consistent parts of information with equivalent ones should not change the inconsistency value [39]. Inconsistency measures that satisfy these principles, either respect the semantics σ of the argument graph, or are so broad s.t. they do not take the semantics into account. Measures that violate these postulates indicate that they tend to focus on other semantics or could be considering structural differences.

Principle 10. p-*Exchange*: If $G \equiv_s^p G'$, then $I(G) = I(G')$.

Principle 11. g-*Exchange*: If $G \equiv_s^g G'$, then $I(G) = I(G')$.

Principle 12. s-*Exchange*: If $G \equiv_s^s G'$, then $I(G) = I(G')$.

Principle 13. $naive$-*Exchange*: If $G \equiv_s^{naive} G'$, then $I(G) = I(G')$.

4.1.5 Additivity Postulates

The fourteenth principle, Disjoint Additivity [24] claims that if two graphs G_1 and G_2 are disjoint then the inconsistency of their combined graph, denoted, $G_1 + G_2$ with $+$ indicating their combination as defined in Definition 2.16 in Section 2.2, is the sum of their inconsistency values. This ensures that the the disjoint components can be assessed independently, which simplifies the process of evaluating inconsistency. Moreover, measures that fulfill this principle enable us to understand how different components contribute to the total inconsistency in the overall graph since conflicts in G_1 are quantified separately from conflicts in G_2. Additionally this postulate shows us that the combination of graphs G_1 and G_2 does not create additional conflicts. Inconsistency measures that violate Disjoint Additivity, suggest that the combination of the two graphs, introduces new conflicts or dependencies, which the measures is sensitive to, or that the measure's focus does not allow for an independent treatment of the graphs.

Principle 14. *Disjoint Additivity*: If G_1 and G_2 are disjoint, then $I(G_1 + G_2) = I(G_1) + I(G_2)$.

The fifteenth principle, Super Additivity [24], states that the inconsistency value of a joint graph G_1+G_2 is greater or equal to the addition of the inconsistency values of G_1 and G_2, thereby strengthening the principle of Monotonicity [35]. The principle of Super Additivity acknowledges that new conflicts can arise from interactions between previously separated graphs. This postulate further is relevant because it ensures that the inconsistency measure behaves predictably when argument graphs are combined. If two separate argument graphs are inconsistent, combining them should logically result in a greater or equal level of inconsistency, and this should be reflected in the measure. Thus, measures that fulfill this postulate acknowledge that the union of two graphs might introduce additional inconsistencies beyond those present in the individual graphs.

Principle 15. *Super Additivity*: $I(G_1 + G_2) \geq I(G_1) + I(G_2)$.

4.1.6 Cyclicality Postulates

The sixteenth principle, Reinforcement [1], only is applicable to acyclic graphs that have the same number of nodes and arcs. It states that an acyclic graph that has indirect attacks is considered more conflicting than an acyclic one containing only direct attacks [1]. This is due to the nature of indirect attacks. The chaining of attacks creates a broader conflict because it implies a more complex relationship between arguments. In contrast, direct attacks only represent one-to-one conflicts. Hence, this postulate suggests that not only the structure of the attack relations matters, but the number of attacks as well because even when two graphs have the same number of arguments and the same number of direct attacks, the presence of indirect attacks can intensify the overall conflict. It is apparent that Reinforcement is most relevant when measuring inconsistency in graphs where indirect attacks matter. If a measure satisfies Reinforcement, it tends to capture both direct and indirect conflicts, making it sensitive to structural complexity. If a measure does not satisfy this rationality postulate, it may underestimate inconsistency because it disregards indirect attacks.

Principle 16. *Reinforcement*: For two acyclic argumentation graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}', \mathcal{R}')$ s.t. $\mathcal{A} = \mathcal{A}' = \{a_0, ...a_n, b_0....b_n\}$ with $n \geq 3, \mathcal{R} = \{(a_i, b_i) \mid i \in \{0, ..., n-1\}\}$, and $\mathcal{R}' = \{(a_i, a_{\text{i+1}}) \mid i \in \{0, ..., n-1\}\}$, it holds true that $I(G') > I(G)$.

To exemplify, we present the case, $n = 3$, of two graphs T and T' constructed in accordance with the Reinforcement postulate's criteria in the following example.

Example 4.1 For $T = (\mathcal{A}, \mathcal{R})$ and $T' = (\mathcal{A}', \mathcal{R}')$ to fulfill Reinforcement criteria, they must be acyclic, have the same number of arcs and the same number of nodes, where each graph must have a minimum of eight nodes with $n = 3$, where the nodes of both graphs have to be grouped in $\{a_0, a_1, ..., a_3\}$ and $\{b_0, b_1, ..., b_3\}$. This results in the following nodes for both graphs $\mathcal{A} = \mathcal{A}' = \{a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3\}$. These grouped nodes follow special attack patterns. For T, $\mathcal{R} = \{(a_i, b_i) \mid i \in \{0, ..., 3 - 1\}\} = \{(a_0, b_0), (a_1, b_1), (a_2, b_2)\}$. And for T', $\mathcal{R}' = \{(a_i, a_{i+1}) \mid i \in \{0, ..., 3 - 1\}\}$ the attack pattern makes sure that there exists at least one indirect attack. Thus, the attacks of T' are $\mathcal{R}' = \{(a_0, a_1), (a_1, a_2), (a_2, a_3)\}$, where the indirect attack is (a_0, a_3). Figure 12 shows the constructed graph T and Figure 13 depicts T'.

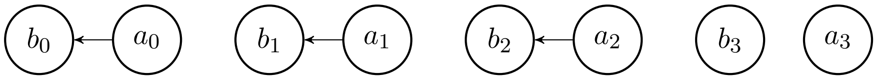


Figure 12: Argument graph T.

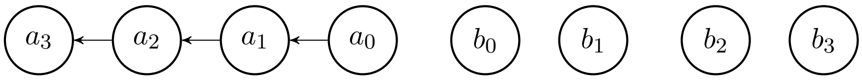


Figure 13: Argument graph T'.

The following principles involve cycles with the seventeenth postulate, Cycle Precedence [1], ensuring that an argumentation graph consisting of an elementary cycle is more inconsistent than an acyclic graph. This is because an overwhelming weight is given to the cycle that represents a deadlock, instead of the inconsistency caused by direct attacks suggesting open conflict [1]. Amgoud and Ben-Naim [1] further illustrate that, according to Cycle Precedence, more weight is given to inconsistency in the cycle s.t. an acyclic graph with 100 direct attacks would be considered less inconsistent than an elementary cycle with 10 attacks. Inconsistency measures that satisfy this postulate assign higher inconsistency value to cycles, which aligns with the intuition that circular arguments are problematic [1] [3]. Measures that do not fulfill this postulate tend not to focus on cycles, meaning they treat cyclic and acyclic argument graphs equally and/or prioritize other aspects.

Principle 17. *Cycle Precedence*: For all argumentation graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}', \mathcal{R}')$, if G is acyclic and G' is an elementary cycle, then $I(G) < I(G')$.

The eighteenth principle, Size Sensitivity [1], proposes that an inconsistency measure could take the sizes of cycles into account where the larger the cycle, the less severe the inconsistency. The idea is that, the less arguments are needed to produce a cycle, the more strong the disagreement [1] and the more difficult it is to resolve conflicts. For instance, a cycle of length 2 is more conflicting than a cycle of length 1000 according to a measure that satisfies this principle [1]. If an inconsistency measures fulfills this postulate it is likely that it is sensitive to the size of cycles. Measures that do not satisfy this principle might not consider cycles or focus on acyclic aspects.

Principle 18. *Size Sensitivity*: For all elementary cycles $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}', \mathcal{R}')$, if $|\mathcal{A}'| < |\mathcal{A}|$, then $I(G) < I(G')$.

4.2 Compliance of Inconsistency Measures with Rationality Postulates

For each inconsistency measure in Section 3, we check whether it satisfies the principles in Section 4.1 or not. For improved readability, the proofs outlined in the following sections are organized according to the sets of postulates and will follow the order of the rationality postulates. For each proof, we first present the inconsistency measures that satisfy the principle, followed by the remaining measures, in the order in which they were proposed.

Table 15 summarizes the results of which principles are satisfied or violated by the inconsistency measures, where for each measure the checkmark (✓) indicates the satisfaction and the cross (✗) symbolizes the violation of the principle. The proofs for the satisfaction or violation of the principles, that were conducted in this thesis are indicated by green and red colored symbols, respectively. Those measures whose compliance has been shown before have been marked correspondingly with [24], [1], [29], and [38], and are not colored. The results in the table are discussed below.

We want to emphasize once more, that we have grouped the postulates based on their structural characteristics for organizational purposes. However, in the discussion of our results below, we deliberately omit this grouping, as it does not directly contribute to the analysis or interpretation of our findings.

Principle	$\mathcal{I}_{\mathrm{pr}}$	$\mathcal{I}_{\mathrm{ngr}}$	$\mathcal{I}_{\mathrm{nst}}$	$\mathcal{I}_{\mathrm{n}}$	$\mathcal{I}_{\mathrm{p}}$	$\mathcal{I}_{\mathrm{dr}}$	$\mathcal{I}_{\mathrm{win}}$	$\mathcal{I}_{\mathrm{wout}}$	$\mathcal{I}_{\mathrm{cc}}$	$\mathcal{I}_{\mathrm{wcc}}$	$\mathcal{I}_{\mathrm{wco}}$	$\mathcal{I}_{\mathrm{con}}$	$\mathcal{I}_{\mathrm{ind}}$	$\mathcal{I}_{\mathrm{db}}$
Consistency	✓[24]	✓[24]	✓[24]	✓[1]	✓[1]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[1]	✓[1]	✓[1]
Normalization	✗	✗	✗	✗	✗	✓[38]	✗	✗	✗	✗	✗	✗	✗	✓
Contradiction	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Monotonicity	✗[24]	✗[24]	✗[24]	✗	✗	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓	✓	✗
Freeness	✓[24]	✓[24]	✓[24]	✓[1]	✓[1]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[1]	✓[1]	✗[3]
Penalty	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓
Free-Node Dilution	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Inversion	✗[24]	✗[24]	✗[24]	✓[29]	✗[29]	✓[24]	✗[24]	✗[24]	✓[24]	✓[24]	✓[24]	✓[29]	✗[29]	✓[29]
Isomorphic Invar.	✓[24]	✓[24]	✓[24]	✓[1]	✓[1]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[1]	✓[1]	✓[1]
p-Exchange	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
g-Exchange	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗
s-Exchange	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
$naive$-Exchange	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
Disjoint Additivity	✗[24]	✓[24]	✓[24]	✗	✗	✗[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓[24]	✓	✓	✗
Super Additivity	✗[24]	✗[24]	✗[24]	✗	✗	✗[24]	✗[24]	✗[24]	✓[24]	✓[24]	✗[24]	✗	✗	✗
Reinforcement	✗	✗	✗	✗[1]	✗[1]	✗	✗	✗	✗	✗	✓	✗[1]	✗[1]	✓[1]
Cycle Precedence	✗	✓	✗	✗[1]	✗[1]	✗	✗	✗	✓	✓	✗	✗[1]	✗[1]	✓[1]
Size Sensitvity	✗	✗	✗	✗[1]	✗[1]	✗	✗	✗	✗	✓	✗	✗[1]	✗[1]	✓[1]

Table 15: Satisfaction (✓) and violation (✗) of the inconsistency measure postulates.

[3] We disprove Amgoud and Ben-Naim's claim in [1], that $\mathcal{I}_{\mathrm{db}}$ satisfies Freeness.

Since Consistency is the only generally accepted postulate and minimal requirement for an inconsistency measure [39], it is satisfied by all inconsistency measures. Normalization, regarded as an optional principle, as many measures tend to assess inconsistency absolutely and not relatively [39] is fulfilled by I_{dr} and I_{db} only. However, it needs to be stated that I_{dr} only adheres to Normalization because of its simplicity as it can only take two values 0 and 1, while I_{db} satisfies Normalization because it is normalized with upper and lower boundaries. In fact, I_{dr} and I_{db} satisfy more postulates than any other inconsistency measure, with I_{dr} complying with 11 out of 18 principles, and I_{db} adhering 10 out of 18 postulates. Yet, the fulfillment of postulates by I_{dr} is based on the coarseness of the measure while the satisfaction of more principles by I_{db} is based on its fine-granular nature. Contradiction, only is satisfied by I_{db}, implying that the Distance-Based Measure is the only inconsistency measure that assigns 1 to a maximally inconsistent argument graph while ensuring that no other argument graph, unless it is maximally inconsistent, can receive a value of 1. We reiterate that Contradiction only makes sense, when Normalization is fulfilled [39] since it specifies the maximum value of Normalization reserved for the complete graph.

Monotonicity, also one of the least disputed postulates in literature [38], is satisfied by all structural inconsistency measures, except for I_{db}. The measure I_{db} is non-monotonic because when new arguments or attacks are added, especially disconnected ones, the overall distance in the graph increases due to the normalization, which may increase the measure. Monotonicity also is violated by all extension-based measures. However, we emphasize that this is due to the inherent non-monotonicity of extension-based semantics, as new information might actually reduce inconsistency by resolving a conflict [4]. Freeness is satisfied by all inconsistency measures which can be attributed to its foundational nature, that irrelevant information should not influence the inconsistency, except for I_{db}. Penalty, complementary to Freeness [35], is satisfied only by I_{con} and I_{db}. This can be attributed to the fact that they consider each arc individually, ensuring that the introduction of a new arc, influences, but does not negate or replace the existence of another. While I_{win} and I_{wout} also directly consider arcs through the examination of in- and out-degrees, respectively, the influence of new arcs may be reduced through weights. Free-Node Dilution, which demands that the addition of an isolated node should not increase the measure, is satisfied by all inconsistency measures.

From the extension-based measures only I_{n} satisfies Inversion, due to its straightforward counting mechanism and simplicity as one of the weakest extensions along with conflict-freeness [14]. From the structure-based measures only I_{win}, I_{wout}, and I_{ind} do not satisfy Inversion. This is related to the fact that they all consider the (weighted) in- or out-degrees which naturally change with Inversion. All inconsistency measures adhere to the principle of Isomorphic Invariance, which means that these measures are not dependent on specific labels or representations.

All postulates related to strong equivalence are satisfied by I_{dr} and I_{n}. For I_{dr} this is once more attributed to its uncomplicated nature. In contrast, I_{n} adheres to the strong equivalence postulates because it focuses on maximal (w.r.t. $\subseteq$) conflict-free extensions which are always preserved under strong equivalence [30][19]. Since p-Exchange preserves preferred extensions, it is satisfied by I_{pr} and I_{p}. Similarly, I_{ngr} complies with g-Exchange based on the maintenance of the grounded extension. The same holds for I_{nst} w.r.t s-Exchange. Counter-intuitively, I_{nst} also complies with the p-Exchange because under strong equivalence w.r.t. preferred semantics, the graphs must not only have the same preferred extensions but also require the same minimal modifications to achieve stability. Other than I_{dr}, most structural measures do not satisfy the strong equivalence postulates. The exceptions are I_{wco} and I_{ind}. WeightedComponentCount conforms to $naive$-Exchange based on the fact that the criteria for strong equivalence under naive semantics conserves the components, their quantity and the included arguments. The In-degree Measure, which quantifies the attacked arguments, honors g-Exchange based on the preservation of attack relations w.r.t. the grounded extension.

The only structure-based measures which do not adhere to Disjoint Additivity are I_{dr}, which can be traced back to its binary nature, and I_{db}, where with the partition or addition of disjoint graphs new distances are created that are accounted for in the measure. The only extension-based measures that do not violate Disjoint Additivity are I_{ngr} and I_{nst} because they are defined in such a way that when the graph is partitioned into disjoint parts, the extensions themselves split or add accordingly. Super Additivity only is satisfied by I_{cc} and I_{wcc}, which focus on cycles.

I_{db} is the only measure that satisfies all three principles, Reinforcement, Cycle Precedence, and Size Sensitivity. Since Reinforcement deals with acyclic graphs, the only other inconsistency measure that adheres to it is I_{wco}. However, it should be noted that the satisfaction of Reinforcement is also tied to the many restrictions and criteria of Reinforcement. These also are part of why other inconsistency measures do not comply with it. Similarly, I_{cc} and I_{wcc} which focus on cycles, comply with Cycle Precedence, because the principles compares an acyclic and an elementary cycles, s.t. Cycle Precedence holds immediately. Cycle Precedence also is fulfilled by I_{ngr}, the only extension-based measure that fulfills a postulate related to cycles, based on the fact that cycles cause all arguments in the graph to be non-grounded [7]. As for Size Sensitivity, it is satisfied by I_{wcc} since it, along with I_{db}, is the only measure that takes the size of cycles into account.

4.2.1 Basic Postulate Propositions and Proofs

Proposition 4.1 *Normalization* is satisfied by I_{db} but not by I_{pr}, I_{ngr}, I_{nst}, I_{n}, I_{p}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{wco}, I_{con}, and I_{ind}.

Proof.

- **(I_{db})** According to Amgoud and Ben-Naim [1], for an argumentation graph $G = (\mathcal{A}, \mathcal{R})$, global distance $GD(G) = \max$ iff $\mathcal{R} = \emptyset$ and $GD(G) = \min$ iff $\mathcal{R} = \mathcal{A} \times \mathcal{A}$. For $I_{\text{db}}(G) = \frac{\max - GD(G)}{\max - \min}$ the following shows that for its endpoints

$I_{\text{db}} \in [0, 1]$ applies:

$$\text{Maximal } GD(G) : I_{db}(G) = \frac{\max - \max}{\max - \min} = 0$$
$$\text{Minimal } GD(G) : I_{db}(G) = \frac{\max - \min}{\max - \min} = 1$$

Since $GD(G) \in [\min, \max]$, $(\max - GD(G)) \in [\max - \max, \max - \min]$. Given that $\max - \min > 0$, it follows that $I_{\text{db}} = \frac{\max - GD(G)}{\max - \min} \in [0, 1]$.

For the following proofs consider graph O in Figure 8 as well as its semantics in Table 8, unless mentioned otherwise.

- **(I_{pr})** $I_{\text{pr}}(O) = 3 - 1 = 2$ s.t. $I_{\text{pr}} \notin [0, 1]$.
- **(I_{ngr})** Consider argument graph A in Figure 1. As calculated in Example 3.6, $I_{\text{ngr}}(A) = 2$ s.t. $I_{\text{ngr}} \notin [0, 1]$.
- **(I_{nst})** Consider argument graph $H + N$ which consists of graphs H in Figure 3 and N in Figure 8. For graph H consider the semantics in Table 12 and for graph N the semantics are depicted in Table 7. For the induced graph where one node attacks the other, observe the semantics in Table 1, and for an isolated node the semantics are available in Table 6 given their isomorphism.

$$N = \emptyset : \text{Extension}_s(\text{Induced}(H + N, \emptyset)) = \text{Extension}_s(\{e, f, g, a\}) = \emptyset$$
$$N = \{e\} : \text{Extension}_s(\text{Induced}(H + N, \{e\})) = \text{Extension}_s(\{f, g, a\}) = \emptyset$$
$$N = \{f\} : \text{Extension}_s(\text{Induced}(H + N, \{f\})) = \text{Extension}_s(\{e, g, a\}) = \emptyset$$
$$N = \{g\} : \text{Extension}_s(\text{Induced}(H + N, \{g\})) = \text{Extension}_s(\{e, f, a\}) = \emptyset$$
$$N = \{a\} : \text{Extension}_s(\text{Induced}(H + N, \{a\})) = \text{Extension}_s(\{e, f, g\}) = \emptyset$$
$$N = \{e, f\} : \text{Extension}_s(\text{Induced}(H + N, \{e, f\})) = \text{Extension}_s(\{g, a\}) = \emptyset$$
$$N = \{e, g\} : \text{Extension}_s(\text{Induced}(H + N, \{e, g\})) = \text{Extension}_s(\{f, a\}) = \emptyset$$
$$N = \{a, e\} : \text{Extension}_s(\text{Induced}(H + N, \{a, e\})) = \text{Extension}_s(\{f, g\}) = \{f\}$$
$$N = \{f, g\} : \text{Extension}_s(\text{Induced}(H + N, \{f, g\})) = \text{Extension}_s(\{a, e\}) = \emptyset$$
$$N = \{f, a\} : \text{Extension}_s(\text{Induced}(H + N, \{f, a\})) = \text{Extension}_s(\{e, g\}) = \{g\}$$
$$N = \{g, a\} : \text{Extension}_s(\text{Induced}(H + N, \{g, a\})) = \text{Extension}_s(\{f, e\}) = \{e\}$$
$$N = \{e, f, g\} : \text{Extension}_s(\text{Induced}(H + N, \{e, f, g\})) = \text{Extension}_s(\{a\}) = \emptyset$$
$$N = \{e, f, a\} : \text{Extension}_s(\text{Induced}(H + N, \{e, f, a\})) = \text{Extension}_s(\{g\}) = \{g\}$$
$$N = \{f, g, a\} : \text{Extension}_s(\text{Induced}(H + N, \{f, g, a\})) = \text{Extension}_s(\{e\}) = \{e\}$$
$$N = \{e, a, g\} : \text{Extension}_s(\text{Induced}(H + N, \{e, a, g\})) = \text{Extension}_s(\{f\}) = \{f\}$$
$$N = \{e, f, g, a\} : \text{Extension}_s(\text{Induced}(H + N, \{e, f, g, a\})) = \text{Extension}_s(\emptyset) = \emptyset$$

$N = \{a, e\} = \{f, a\} = \{g, a\}$ are the smallest sets that result in stable extensions and $I_{\text{nst}} = |N| = \{a, e\} = \{f, a\} = \{g, a\} = 2$ s.t. $I_{\text{nst}} \notin [0, 1]$.

- **(I_{n})** $I_{\text{n}}(O) = 3 + 0 - 1 = 2$ s.t. $I_{\text{n}} \notin [0, 1]$.
- **(I_{p})** $I_{\text{p}}(O) = 3 + 0 - 1 = 2$ s.t. $I_{\text{p}} \notin [0, 1]$.

Consider graph C in Figure 2 as a counterexample in the subsequent proofs for each measure unless stated otherwise.

- **(I_{win})** $I_{\text{win}}(C) = 1 + 1 + 1 + \frac{1}{2} = \frac{7}{2} = 3.50$ s.t. $I_{\text{win}} \notin [0, 1]$.
- **(I_{wout})** $I_{\text{wout}}(C) = 5 \times 1 = 5$ s.t. $I_{\text{wout}} \notin [0, 1]$.
- **(I_{cc})** For this measure consider graph L in Figure 7. $I_{\text{cc}}(L) = 2$ s.t. $I_{\text{cc}} \notin [0, 1]$.
- **(I_{wcc})** For this measure once more consider graph L in Figure 7. $I_{\text{wcc}}(L) = 1 + \frac{1}{2} = \frac{3}{2} = 1.50$ s.t. $I_{\text{wcc}} \notin [0, 1]$.
- **(I_{wco})** $I_{\text{wco}}(C) = (5 - 1)^2 = 16$ s.t. $I_{\text{wco}} \notin [0, 1]$.
- **(I_{con})** $I_{\text{con}}(C) = 5$ s.t. $I_{\text{con}} \notin [0, 1]$.
- **(I_{ind})** $I_{\text{ind}}(C) = 4$ s.t. $I_{\text{ind}} \notin [0, 1]$. □

Proposition 4.2 *Contradiction* is satisfied by I_{db} but not by I_{pr}, I_{ngr}, I_{nst}, I_{n}, I_{p}, I_{dr}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{wco}, I_{con}, and I_{ind}.

Proof.

- **(I_{db})** In the following we prove that $I_{\text{db}} = 1 \Leftrightarrow \forall \mathcal{S} \subseteq \mathcal{A},\ \mathcal{S} \neq \emptyset$, and $\mathcal{S} \models \perp$ holds in both directions.
  - $\Leftarrow$ Recall that Contradiction requires that every possible non-empty subset $\mathcal{S}$ of arguments in the argumentation graph is inconsistent. The smallest non-empty subset in a graph would be single nodes. For each and every one of them to be considered inconsistent, they have to be self-attacking. For every subset to be considered inconsistent there needs to be attacks between every pair of nodes as well. The result is a complete graph. For a complete graph $G = (\mathcal{A}, \mathcal{R})$ with $n = |\mathcal{A}|$ where $n \geq 1$ it holds that $|\mathcal{R}| = n^2$ by definition. It follows that $GD(G) = n^2$, and $max = n^2 \times (n+1)$ and $min = n^2$. Thus it holds that

$$I_{db}(G) = \frac{\max - GD(G)}{\max - \min} = \frac{n^2 \times (n+1) - n^2}{n^2 \times (n+1) - n^2} = 1.$$

  - $\Rightarrow$ If for a graph $G = (\mathcal{A}, \mathcal{R})$, $I_{\text{db}}(G) = 1$ then, $\max - GD(G) = \max - \min$ because $\max$ and $\min$ are constants. Therefore it must hold that $GD(G) = \min$. According to [1] the global distance is at a minimum iff $\mathcal{R} = \mathcal{A} \times \mathcal{A}$ which only is the case in a complete graph.

Unless mentioned otherwise, for the following proofs consider argument graph E in Figure 3 and its semantics in Table 9. Every possible set of arguments, $\{x\}$, $\{y\}$, and $\{x, y\}$ is inconsistent, thus E satisfies the conditions of Contradiction and should receive an inconsistency measure of 1. The following counterexamples show that this is not the case.

- **(I_{pr})** The following counterexamples show that I_{pr} violates Contradiction.

  $\Leftarrow$ $I_{\text{pr}}(E) = 1 - 1 = 0 \neq 1$.

  $\Rightarrow$ Consider the semantics of graph A from Figure 1 in Table 10. $I_{\text{pr}}(A) = 2 - 1 = 1$ but argument graph A is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{ngr})** In the following we prove that I_{ngr} does not satisfy Contradiction.

  $\Leftarrow$

  $$\begin{aligned} I_{ngr}(E) &= |\text{Nodes}(E) \setminus (\text{Extension}_g(E) \cup \text{Attacked}(E))| \\ &= |\{x, y\} \setminus (\emptyset \cup \emptyset)| \\ &= |\{x, y\} \setminus \emptyset| = |\{x, y\}| = 2 \neq 1. \end{aligned}$$

  $\Rightarrow$ Consider the graph $B + N$ consisting of two disjoint components, B in Figure 2 and N in Figure 8, and their Semantics in Table 1 and 7, respectively.

  $$\begin{aligned} I_{ngr}(B + N) &= |\text{Nodes}(B + N) \setminus (\text{Extension}_g(B + N) \cup \text{Attacked}(B + N))| \\ &= |\{b, c, a\} \setminus (\{c\} \cup \{b\})| \\ &= |\{b, c, a\} \setminus \{c, b\}| = |\{a\}| = 1. \end{aligned}$$

  $I_{\text{ngr}}(B + N) = 1$ but $B + N$ is not a complete graph.

- **(I_{nst})** The following counterexamples show that I_{nst} violates Contradiction.

  $\Leftarrow$

  $$\begin{aligned} N = \emptyset &: \text{Extension}_s(\text{Induced}(E, \emptyset)) = \text{Extension}_s(\{x, y\}) = \emptyset \\ N = \{x\} &: \text{Extension}_s(\text{Induced}(E, \{x\})) = \text{Extension}_s(\{y\}) = \emptyset \\ N = \{y\} &: \text{Extension}_s(\text{Induced}(E, \{y\})) = \text{Extension}_s(\{x\}) = \emptyset \\ N = \{x, y\} &: \text{Extension}_s(\text{Induced}(E, \{x, y\})) = \text{Extension}_s(\emptyset) = \emptyset \end{aligned}$$

  Since all arguments have to be removed, $I_{\text{nst}}(E) = 2 \neq 1$.

  $\Rightarrow$ As calculated in Example 3.7, $I_{\text{nst}}(P) = 1$ but argument graph P in Figure 9 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{n})** The following counterexamples show that I_{n} violates Contradiction.

  $\Leftarrow$ $I_{\text{n}}(E) = 1 + 2 - 1 = 2 \neq 1$.

- ⇒ As visible from the semantics in Table 10, $I_{\text{n}}(A) = 2 + 0 - 1 = 1$ but argument graph A in Figure 1 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{p})** The following counterexamples show that I_{p} violates Contradiction.
  - ⇐ $I_{\text{p}}(E) = 1 + 2 - 1 = 2 \neq 1$.
  - ⇒ As observable from the semantics in Table 10, $I_{\text{p}}(A) = 2 + 0 - 1 = 1$ but argument graph A in Figure 1 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{dr})** In the following we prove that $I_{\text{dr}} = 1 \Leftrightarrow \forall \mathcal{S} \subseteq \mathcal{A},\ \mathcal{S} \neq \emptyset$, and $\mathcal{S} \models \perp$ only holds in the direction ⇐. Therefore, I_{dr} does not satisfy Contradiction.
  - ⇐ For any complete graph $G = (\mathcal{A}, \mathcal{R})$, $\mathcal{R} \neq \emptyset$ by definition. It follows that $I_{\text{dr}}(G) = 1$.
  - ⇒ For graph B in Figure 2, $I_{\text{dr}}(B) = 1$. Yet, B is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{win})** In the following we prove that $I_{\text{win}} = 1 \Leftrightarrow \forall \mathcal{S} \subseteq \mathcal{A},\ \mathcal{S} \neq \emptyset$, and $\mathcal{S} \models \perp$ only holds in the direction ⇐. Therefore, I_{win} does not satisfy Contradiction.
  - ⇐ Suppose an arbitrary complete graph G has n nodes where $n \geq 1$. Since G is a complete graph every node is attacked by every node including itself. Thus, for every node a in the complete graph G, Indegree(a, G) = n s.t.
  $$I_{\text{win}}(G) = \sum_{a \in \text{Nodes}(G) \text{ s.t. } \text{Indegree}(G,a) \geq 1} \frac{1}{\text{Indegree}(G, a)}$$
  $$I_{\text{win}}(G) = n \times \frac{1}{n} = 1.$$
  - ⇒ $I_{\text{win}}(B) = 1$ but argument graph B in Figure 2 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{wout})** In the following we prove that $I_{\text{wout}} = 1 \Leftrightarrow \forall \mathcal{S} \subseteq \mathcal{A},\ \mathcal{S} \neq \emptyset$, and $\mathcal{S} \models \perp$ only holds in the direction ⇐. Therefore I_{wout} does not satisfy Contradiction.
  - ⇐ Suppose an arbitrary complete graph G has n nodes where $n \geq 1$. Since G is a complete graph every node attacks every node including itself. Thus, for a node a in the complete graph G, Outdegree(a, G) = n, s.t.
  $$I_{\text{wout}}(G) = \sum_{a \in \text{Nodes}(G) \text{ s.t. } \text{Outdegree}(G,a) \geq 1} \frac{1}{\text{Outdegree}(G, a)}$$
  $$I_{\text{wout}}(G) = n \times \frac{1}{n} = 1.$$
  - ⇒ $I_{\text{wout}}(B) = 1$ but argument graph B in Figure 2 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{cc})** The following counterexamples show that I_{cc} violates Contradiction.
  - $\Leftarrow$ $I_{\text{cc}}(E) = 3 \neq 1$.
  - $\Rightarrow$ $I_{\text{cc}}(A) = 1$ but argument graph A in Figure 1 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{wcc})** The following counterexamples show that I_{wcc} violates Contradiction.
  - $\Leftarrow$ $I_{\text{wcc}}(E) = 1 + 1 + \frac{1}{2} = \frac{5}{2} = 2.50 \neq 1$.
  - $\Rightarrow$ $I_{\text{wcc}}(P) = 1$ but argument graph P in Figure 9 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

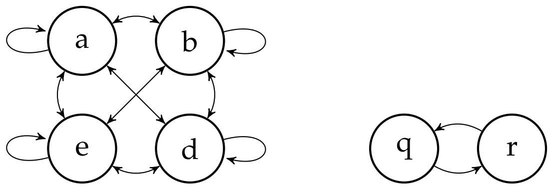


Figure 14: Argument graphs U (left) and V (right).

- **(I_{wco})** The following counterexamples show that I_{wco} violates Contradiction.
  - $\Leftarrow$ Consider complete graph U in Figure 14. $I_{\text{wco}}(U) = (4-1)^2 = 9 \neq 1$.
  - $\Rightarrow$ $I_{\text{wco}}(B) = (2-1)^2 = 1$ but argument graph B in Figure 2 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{con})** The following counterexamples show that I_{con} violates Contradiction.
  - $\Leftarrow$ $I_{\text{con}}(E) = 4 \neq 1$.
  - $\Rightarrow$ $I_{\text{con}}(B) = 1$ but argument graph B in Figure 2 is not a complete graph and therefore does not fulfill the conditions of Contradiction.

- **(I_{ind})** The following counterexamples show that I_{ind} violates Contradiction.
  - $\Leftarrow$ $I_{\text{ind}}(E) = 2 \neq 1$.
  - $\Rightarrow$ $I_{\text{ind}}(B) = 1$ but argument graph B in Figure 2 is not a complete graph and therefore does not fulfill the conditions of Contradiction. □

4.2.2 Expansion Postulate Propositions and Proofs

Proposition 4.3 *Montonicity* is satisfied by I_{con} and I_{ind} but not by I_{n}, I_{p}, and I_{db}.

Proof.

- **(I_{con})** Recall that for a graph $G = (\mathcal{A}, \mathcal{R})$, $I_{\text{con}}(G) = |\mathcal{R}|$. For $G \subseteq G'$, it holds by definition that $\mathcal{A} \subseteq \mathcal{A}'$ and $\mathcal{R} \subseteq \mathcal{R}'$. It follows automatically that $I_{\text{con}}(G) = |\mathcal{R}| \leq |\mathcal{R}'| = I_{\text{con}}(G')$.
- **(I_{ind})** Recall that the In-degree Measure quantifies the number of attacked arguments. It holds that $G \subseteq G'$ s.t. if we add arcs and/or nodes to G, we construct G'. We distinguish between the following cases.
  - We add arguments to G to construct G':
    * If the new argument is not attacked I_{ind} remains unchanged.
    * If the new argument is attacked, I_{ind} increases.
  - We add arcs to G to construct G':
    * If the arc targets an argument that is already attacked I_{ind} remains the same.
    * If the arc newly attacks an argument I_{ind} increases.

  In all cases it holds that $I_{\text{ind}}(G) \leq I_{\text{ind}}(G')$.
- **(I_{n})** Consider an argument graph consisting of the components V in Figure 14 and C' in Figure 3, $V + C'$ and another graph, $V + P$ consisting of the graphs V in Figure 14 and P from 9. The newly formed graph $V + C'$ has four naive extensions $\{r, a\}, \{r, b\}, \{q, a\}, \{q, b\}$ and no self-attacks. In contrast, $V + P$ has two naive extensions $\{a, q\}, \{a, r\}$ and one self-attack. It is visible that $I_{\text{p}}(V + C') = 4 + 0 - 1 = 3 \nleq 2 = 2 + 1 - 1 = I_{\text{p}}(V + C')$ although $V + C' \subseteq V + P$.
- **(I_{p})** Consider graphs A in Figure 1 and D in Figure 2 and their semantics in Table 10 and 16. $A \subseteq D$ but $I_{\text{p}}(A) = 2 + 0 - 1 = 1 \nleq 0 = 1 + 0 - 1 = I_{\text{p}}(D)$.
- **(I_{db})** In the proof for Freeness under Proposition 4.4, below, we calculate $I_{\text{db}}(V) = 0.75 \leq 0.37 \approx I_{\text{db}}(V')$ although $V \subseteq V'$.

Set(D)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
∅	✓	✓	✗	✗	✗	✗	✗
{a}	✓	✓	✗	✗	✗	✗	✗
{b}	✓	✗	✗	✗	✗	✗	✓
{c}	✓	✓	✗	✗	✗	✗	✗
{a, c}	✓	✓	✓	✓	✓	✓	✓

Table 16: Semantics of argument graph D.

Proposition 4.4 *Freeness* is not satisfied by I_{db}.

Proof.

- **(I_{db})** As a counterexample, consider graph V in Figure 14 with $I_{\text{db}}(V) = 0.75$ as calculated in Example 3.28 with isomorphic graph A in Figure 1. Suppose a disconnected new node a is added to V. Let $V \cup a$ be denoted as V'. Graph V' now has $n = 3$ nodes s.t. $\max = n^2 \times (n+1) = 3^2 \times (3+1) = 36$ and $\min = n^2 = 3^2 = 9$. The global distance of the new graph is $GD(V') = 2 \times (1+2+4) + (3 \times 4) = 26$.

$$I_{db}(V') = \frac{\max - GD(V')}{\max - \min} = \frac{36-26}{36-9} = \frac{10}{27} \approx 0.37.$$

  Nodes(V) = Nodes(V') $\setminus \{a\}$ and Arcs(V) = Arcs(V'), but $I_{\text{db}}(V) = 0.75 \neq 0.37 \approx I_{\text{db}}(V')$. □

Proposition 4.5 *Penalty* is satisfied by I_{con} and I_{db} but not by I_{pr}, I_{ngr}, I_{nst}, I_{n}, I_{p}, I_{dr}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{wco}, and I_{ind}.

Proof.

- **(I_{con})** For an arbitrary graph $G = (\mathcal{A}, \mathcal{R})$ with $|\mathcal{R}| = n$ arcs, $I_{\text{con}}(G) = n$ where $n \in \mathbb{Z}, n \geq 0$. If an arc r is added that attacks any node in G, $I_{\text{con}}(G + \{r\}) = n + 1 > n = I_{\text{con}}(G)$.

- **(I_{db})** Recall the global distance $GD(G)$ is the sum of lengths of the shortest paths between any pair of arguments in the argument graph which is calculated based on the following rules.

  1. If node b is reachable from node a then $d(a, b)$ is the shortest distance between them, the smallest number of arcs from node a to b.
  2. If node b is not reachable from a, then $d(a, b) = k = |\mathcal{A}| + 1$.
  3. If $a = b$, then $d(a, b)$ is the length of the shortest elementary cycles in which a is involved in. If a is not be involved in any cycles, $d(a, b) = k$.

  Observe an arbitrary graph $G = (\mathcal{A}, \mathcal{R})$ where an arbitrary arc $r = (x, y)$ is added, resulting in the new graph $G + \{r\}$. We distinguish between the following cases:

  – Node y was not reachable from x s.t. $d(x, y) = k$ where $k \in \mathbb{Z}, k > 1$. With the added arc r, y becomes reachable for x and now it holds that $d(x, y) = 1$. This results in $GD(G + \{r\}) = GD(G) - k + 1$. If for other nodes, the additional arc adds a shorter path, it may be that their distance decrease as well. It applies that $GD(G) > GD(G + \{r\})$.

– Node y was already reachable for x with a path of length $p \in \mathbb{Z}, p > 1$ resulting in $d(x, y) = p$. However, with the added arc (x, y) a shorter path with length 1 now exists. This decreases the global distance by p and adds 1 to it. The new global distance is $GD(G + \{r\}) = GD(G) - p + 1$. Once again, it is also possible that through r other paths are shortened, further decreasing the global distance. $GD(G) > GD(G + \{r\})$ holds.

– Adding $r = (x, y)$ closes a cycle of length s where $1 < s < k$. In this case for every node involved in the cycle, $d(x, y)$ where $x = y$, will be reduced from k to the length of the elementary cycle s. The result will be a decreased global distance $GD(G+\{r\}) = GD(G) - (s \times k) + (s \times s)$. As previously stated, if other paths are affected by this added arc, the global distance may even decrease more. $GD(G) > GD(G + \{r\})$ holds.

– A self-attack $r = (x, y)$ where $x = y$ is added. Before the self-attack, $d(x, x) = k$ if it is not involved in any elementary cycle, and $d(x, x) = s$ where $1 < s < k$ is the length of the elementary cycle x is involved in. After the added arc, the distance decreases to $d(x, x) = 1$. With that the global distance has either decreased by k or s and increased by 1. The new global distance therefore would be $GD(G + \{r\}) = GD(G) - k + 1$ or $GD(G + \{r\}) = GD(G) - s + 1$. In either case it holds that $GD(G) > GD(G + \{r\})$ holds.

As visible, the addition of an arbitrary arc r always shortens at least one path. Since the number of nodes $n = |\mathcal{A}|$ is constant, $max = n^2 \times (n + 1)$ and $min = n^2$ also remain constant. It follows that the only thing that changes in the distance based measure is the global distance. As we have shown $GD(G) > GD(G + \{r\})$ holds. Therefore it also holds that

$$I_{db}(G) = \frac{\max - GD(G)}{\max - \min} < \frac{\max - GD(G + \{r\})}{\max - \min} = I_{db}(G + \{r\}).$$

- **(I_{pr})** Analyze argument graph M in Figure 8 and its semantics in Table 6. $I_{\text{pr}}(M) = 1 - 1 = 0$. Suppose an arc $r = (x, x)$ is added. The resulting graph is isomorphic to graph N in Figure 8 whose semantics are in Table 7. $I_{\text{pr}}(M + \{r\}) = 1 - 1 = 0$. It is visible that $I_{\text{pr}}(M) \nless I_{\text{pr}}(M + \{r\})$.

- **(I_{ngr})** Analyze graph A in Figure 1. As calculated in Example 3.6, $I_{\text{ngr}}(A) = 2$. Suppose an arc $r = (a, a)$ is added to graph A, with the resulting graph $I_{\text{ngr}}(A + \{r\})$ equating graph L in Figure 7 whose semantics are in Table 17.

$$\begin{aligned} I_{ngr}(A + \{r\}) &= |\text{Nodes}(L) \setminus (\text{Extension}_g(L) \cup \text{Attacked}(L)| \\ &= |\{b, a\} \setminus (\emptyset \cup \emptyset)| \\ &= |\{b, a\} \setminus \emptyset| = |\{a, b\}| = 2. \end{aligned}$$

Thus, $I_{\text{ngr}}(A) = 2 \nless 2 = I_{\text{ngr}}(A + \{r\})$.

Set(L)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	✓	✗	✓	✗	✗
{a}	✗	✗	✗	✗	✗	✗	✗
{b}	✓	✓	✓	✓	✗	✓	✓

Table 17: Semantics of argument graph L.

- **(I_{nst})** Observe argument graph B in Figure 2. As calculated in Example 3.8, $I_{\text{nst}}(B) = 0$. Let graph $B + \{r\}$ be graph B where an arc $r = (b, b)$ is added. The resulting graph is isomorphic to graph L' in Figure 7 whose semantics are in Table 3.

  $$N = \emptyset : \text{Extension}_s(\text{Induced}(B + \{r\}, \emptyset)) = \text{Extension}_s(\{b, c\}) = \{c\}$$

  Since $N = \emptyset$ already has a stable extension $\{c\}$ it holds that $I_{\text{nst}}(B + \{r\}) = 0$. Consequently, $I_{\text{nst}}(B) = 0 \not< 0 = I_{\text{nst}}(B + \{r\})$.

- **(I_{n})** For argument graph B in Figure 2 according to its semantics in Table 1, $I_{\text{n}}(B) = 2 + 0 - 1 = 1$. Assume an arc $r = (b, b)$ is added. The resulting graph $B + \{r\}$ is isomorphic to graph L' whose semantics are available in Table 3. Since $B + \{r\}$ now also a self-attack, $I_{\text{n}}(B + \{r\}) = 1 + 1 - 1 = 1$. $I_{\text{n}}(B) = 1 \not< 1 = I_{\text{n}}(B + \{r\})$.

- **(I_{p})** Consider argument graph L' in Figure 7 and its semantics in Table 3. $I_{\text{pr}}(L') = 1 + 1 - 1 = 1$. Suppose an arc $r = (a, b)$ is added. For the resulting graph $L' + \{r\}$, which is identical to graph L in the same Figure whose semantics are in Table 17, the following holds: $I_{\text{pr}}(L' + \{r\}) = 1 + 1 - 1 = 1$. $I_{\text{pr}}(L') = 1 \not< 1 = I_{\text{pr}}(L' + \{r\})$.

- **(I_{dr})** Examine argument graph B in Figure 2 as a counterexample. $I_{\text{dr}}(B) = 1$. After the addition of one arc $r = (b, c)$, it remains that $I_{\text{dr}}(B + \{r\}) = 1$. As such, $I_{\text{dr}}(B) = 1 \not< 1 = I_{\text{dr}}(B + \{r\})$.

- **(I_{win})** Analyze graph K in Figure 5. $I_{\text{win}}(K) = 1 + \frac{1}{2} = \frac{3}{2} = 1.50$. If an arc $r = (d, a)$ is added, $I_{\text{win}}(K + \{r\}) = 1 + \frac{1}{3} = \frac{4}{3} \approx 1.33$. $I_{\text{win}}(K) = 1.50 \not< 1.33 \approx I_{\text{win}}(K + \{r\})$.

- **(I_{wout})** Once again, observe graph K in Figure 5. $I_{\text{wout}}(K) = 3 \times 1 = 3$. If an arc $r = (c, d)$ is added then $I_{\text{wout}}(K + \{r\}) = 1 + 1 + \frac{1}{2} = \frac{5}{2} = 2.50$. $I_{\text{win}}(K) = 3 \not< 2.50 = I_{\text{win}}(K + \{r\})$.

- **(I_{cc} and I_{wcc})** Take into account graph K in Figure 5 as a counterexample once more. $I_{\text{cc}}(K) = I_{\text{wcc}}(K) = 0$. If an arc $r = (c, d)$ is added, $I_{\text{cc}}(K + \{r\}) = I_{\text{wcc}}(K + \{r\}) = 0$. It follows, $I_{\text{cc}}(K) = 0 \not< 0 = I_{\text{cc}}(K + \{r\})$ and $I_{\text{wcc}}(K) = 0 \not< 0 = I_{\text{wcc}}(K + \{r\})$.

- **(I_{wco})** See graph K in Figure 5 yet again. As calculated in Example 3.22, $I_{\text{wco}}(K) = 5$. If an arc $r = (c, b)$ is added to K, $I_{\text{wco}}(K + \{r\}) = (3-1)^2 + (2-1)^2 = 5$ and $I_{\text{wco}}(K) = 5 \not< 5 = I_{\text{wco}}(K + \{r\})$.

- **(I_{ind})** Consider graph B in Figure 2. $I_{\text{ind}}(B) = 1$. When an arc $r = (b, b)$ is added, $I_{\text{ind}}(B + \{r\}) = 1$, s.t. $I_{\text{ind}}(B) = 1 \not< 1 = I_{\text{ind}}(B + \{r\})$. □

Proposition 4.6 *Free-Node Dilution* is satisfied by I_{pr}, I_{ngr}, I_{nst}, I_{n}, I_{p}, I_{dr}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{wco}, I_{con}, I_{ind}, and I_{db}.

Proof.

- **(I_{pr})** According to Collary 12 from Dung [17] every argumentation graph has at least one preferred extension. Consider a graph G that has n preferred extensions, Extension$_p$(G), where $n \in \mathbb{Z}, n \geq 1$. $I_{\text{pr}}(G) = n - 1$. Suppose a new disconnected node a is added to G and a new graph $G \cup \{a\}$ is formed. Then each existing preferred extension in G, is expanded by $\{a\}$ resulting in the following n extensions: Extension$_p$(G) $\cup \{a\}$. Therefore the number of preferred extensions remains the same, and it holds that $I_{\text{pr}}(G) = I_{\text{pr}}(G \cup \{a\}) = n - 1$. Thus, $I_{\text{pr}}(G) \geq I_{\text{pr}}(G \cup \{a\})$.

- **(I_{ngr})** According to Hunter [24], I_{ngr} satisfies Disjoint Additivity. Suppose for a graph G, $I_{\text{ngr}}(G) = n$ where $n \in \mathbb{Z}, n \geq 0$. For a disconnected new node a, it always holds that $I_{\text{ngr}}((\{a\}, \emptyset)) = 0$.

$$
\begin{aligned}
I_{ngr}(G) &= |\text{Nodes}(G) \setminus (\text{Extension}_g(G) \cup \text{Attacked}(G)| \\
&= |\{a\} \setminus (a \cup \emptyset)| \\
&= |\{a\} \setminus \{a\}| = |\emptyset| = 0
\end{aligned}
$$

  Since disjoint additivity applies it is valid to claim $I_{\text{ngr}}(G + \{a\}) = I_{\text{ngr}}(G) + I_{\text{ngr}}(\{a\}) = n + 0 = n$. Thus, $I_{\text{ngr}}(G) \geq I_{\text{ngr}}(G \cup \{a\})$ holds.

- **(I_{nst})** Hunter [24] has proven that Disjoint Additivty is met by I_{nst}. Thus, when a disconnected new node a is added to an arbitrary graph G, it follows that $I_{\text{nst}}(G + \{a\}) = I_{\text{nst}}(G) + I_{\text{nst}}((\{a\}, \emptyset))$. Suppose $I_{\text{nst}}(G) = n$ where $n \in \mathbb{Z}, n \geq 0$. For a single node a it holds that $I_{\text{nst}}((\{a\}, \emptyset)) = 0$, since $N = \emptyset$ already results in a stable extension.

$$
N = \emptyset : \text{Extension}_s(\text{Induced}(\{a\}, \emptyset)) = \text{Extension}_s(\{a\}) = \{a\}
$$

  Thus, $I_{\text{nst}}(G + \{a\}) = I_{\text{nst}}(G) = n$ and $I_{\text{nst}}(G) \geq I_{\text{nst}}(G \cup \{a\})$ is satisfied.

- **(I_{n})** Consider an arbitrary graph G that has n naive extensions where $n \in \mathbb{Z}, n \geq 1$ and m self attacks where $m \in \mathbb{Z}, m \geq 0$. $I_{\text{n}}(G) = n + m - 1$. Suppose a new node a is added to G. Node a is disconnected and by definition conflict-free and therefore does not introduce any new conflicts. Each existing naive extension in G, Extension$_n(G)$, is expanded by a resulting in Extension$_n(G)$ $\cup$ $\{a\}$. Therefore the number of naive extensions remains the same, and as the self-attacks remain constant, it holds that $I_{\mathrm{n}}(G \cup \{a\}) = n + m - 1$ and $I_{\mathrm{n}}(G) \geq I_{\mathrm{n}}(G \cup \{a\})$.

- **(I_{p})** For I_{pr} we have already established that $I_{\mathrm{pr}}(G) \leq I_{\mathrm{pr}}(G \cup \{a\})$ holds further above. Recall that $I_{\mathrm{p}} = I_{\mathrm{pr}} + |\text{selfAttacks}(G)|$ holds. Since after, the addition of an arbitrary new disconnected node a the number of self-attacks in the graph G remains constant, $I_{\mathrm{p}}(G) \geq I_{\mathrm{p}}(G \cup \{a\})$ holds by extension.

- **(I_{dr})** For an arbitrary graph $G = (\mathcal{A}, \mathcal{R})$, there exists two cases in the Drastic measure:
  - $\mathcal{R} \neq \emptyset$ s.t. $I_{\mathrm{dr}}(G) = 1$. If a new disconnected node a is added, it does not remove or add to the arcs in the graph and it remains that $\mathcal{R} \neq \emptyset$. Hence, $I_{\mathrm{dr}}(G \cup \{a\}) = 1$.
  - $\mathcal{R} = \emptyset$ s.t. $I_{\mathrm{dr}}(G) = 0$. If a new disconnected node a is added, it does not add or remove any arc to the graph. It therefore holds that $\mathcal{R} = \emptyset$ and $I_{\mathrm{dr}}(G \cup \{a\}) = 0$.

  We derive that $I_{\mathrm{dr}}(G) \geq I_{\mathrm{dr}}(G \cup \{a\})$ holds for all cases.

- **(I_{win})** Suppose a disconnected new node a is added to a graph G with $I_{\mathrm{win}}(G) = n$ where $n \in \mathbb{R}$ and $n > 0$. Since argument a has no attacks it does not contribute to $I_{\mathrm{win}}(G)$ s.t. $I_{\mathrm{win}}(G \cup \{a\}) = n$ remains unchanged. $I_{\mathrm{win}}(G) \geq I_{\mathrm{win}}(G \cup \{a\})$ holds.

- **(I_{wout})** Similar to I_{win}, the addition of a new node a has no outgoing attacks s.t. it does not contribute to $I_{\mathrm{wout}}(G) = n$ where $n \in \mathbb{R}$ and $n > 0$. It follows that $I_{\mathrm{wout}}(G + \{a\}) = n$. As a result $I_{\mathrm{wout}}(G) \geq I_{\mathrm{wout}}(G \cup \{a\})$ is satisfied.

- **(I_{cc})** When a disconnected new node a is added to an arbitrary graph G with n cycles where $n \in \mathbb{Z}, n \geq 0$, it will not introduce or remove any new cycles to graph, as it does not introduce or remove any arcs that are part of the cycle, nor creates a new cycle. Hence, $I_{\mathrm{cc}}(G) = I_{\mathrm{cc}}(G \cup \{a\}) = n$ and $I_{\mathrm{cc}}(G) \geq I_{\mathrm{cc}}(G \cup \{a\})$.

- **(I_{wcc})** We differentiate between two cases:
  - Suppose an arbitrary graph G no cycle. Then, $I_{\mathrm{wcc}}(G) = 0$ and it remains $I_{\mathrm{wcc}}(G \cup \{a\}) = 0$ if a disconnected node a is added to G, as it does not create a cycle.
  - Now assume that G has n cycles where $n \in \mathbb{Z}, n \geq 1$, and each cycle C_i where $i = \{1, 2, ..., n\}$ involves $m = |C_i|$ nodes where $m \in \mathbb{Z}, n \geq 1$. If a disconnected new node a is added to G, it will not modify the number of

cycles, the number of nodes involved in a cycle nor create new cycles s.t. $G \cup \{a\}$ will still have n cycles and m nodes involved in these cycles.

$$\begin{aligned} I_{\text{wcc}}(G) &= \sum_{C \in \text{Cycles}(G)} \frac{1}{|C|} \\ &= \sum_{i=1}^{n} \frac{1}{|C_i|} \qquad = I_{\text{wcc}}(G \cup \{a\}) \end{aligned}$$

Hence, $I_{\text{wcc}}(G) \geq I_{\text{wcc}}(G \cup \{a\})$ is proven.

- **(I_{wco})** Recall that

$$I_{wco}(G) = \sum_{X \in \text{Components}(G)} (|X| - 1)^2.$$

If for an arbitrary graph G, for which $I_{\text{wco}}(G) = n$ where $n \in \mathbb{Z}, n \geq 0$, a disconnected new node a is added, a will represent a new component. But since $I_{\text{wco}}(\{a\}) = (1-1)^2 = 0$, a will not contribute to $I_{\text{wco}}(G)$ s.t. $I_{\text{wco}}(G) = I_{\text{wco}}(G \cup \{a\}) = n$ will hold. It applies that $I_{\text{wco}}(G) \geq I_{\text{wco}}(G \cup \{a\})$.

- **(I_{con})** Suppose a graph $G = (\mathcal{A}, \mathcal{R})$ has $n = |\mathcal{R}|$ arcs where $n \in \mathbb{Z}, n \geq 0$. $I_{\text{con}}(G) = n$ as $I_{\text{con}} = |\mathcal{R}|$. Suppose a new disconnected node a is added to graph G forming the graph $G \cup \{a\} = (\mathcal{A} \cup \{a\}, \mathcal{R})$. Since the addition of the node does not modify the arcs $\mathcal{R}$, $I_{\text{con}}(G) = I_{\text{con}}(G \cup \{a\}) = n$ and $I_{\text{con}}(G) \geq I_{\text{con}}(G \cup \{a\})$ is satisfied.

- **(I_{ind})** Visualize an arbitrary graph $G = (\mathcal{A}, \mathcal{R})$ with n attacked nodes where $n \in \mathbb{Z}, n \geq 0$. Since I_{ind} counts the number of attacked nodes in the graph, $I_{\text{ind}}(G) = n$. The addition of a disconnected new node a in G, does not modify the attacks present s.t. $G \cup \{a\} = (\mathcal{A} \cup \{a\}, \mathcal{R})$. Therefore, it remains that $I_{\text{ind}}(G) = I_{\text{ind}}(G \cup \{a\}) = n$. $I_{\text{ind}}(G) \geq I_{\text{ind}}(G \cup \{a\})$ holds.

- **(I_{db})** We refer to the proof of Amgoud and Ben-Naim in [1] under Theorem 6 as it is applicable here.

4.2.3 Strong Equivalence Postulate Propositions and Proofs

Proposition 4.7 *p-Exchange* is satisfied by I_{pr}, I_{nst}, I_{n}, I_{p}, and I_{dr}. This postulate is not satisfied by I_{ngr}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{wco}, I_{con}, I_{ind}, and I_{db}.

Proof.

- **(I_{pr})** Two argument graphs G and G' are strongly equivalent to each other under preferred semantics iff $G^{p\mathcal{K}} = G'^{p\mathcal{K}}$. The p-kernel only removes attacks unrelated to the preferred extensions, ensuring that the $\text{Extension}_p(G)$ = $\text{Extension}_p(G')$. Since $I_{\text{pr}} = |\text{Extension}_p(G')| - 1$, it follows that if $G \equiv_s^p G'$ then $I_{\text{pr}}(G) = I_{\text{pr}}(G')$.

- **(I_{nst})** Consider two graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \mathcal{R}')$ s.t. $G^{p\mathcal{K}} = G'^{p\mathcal{K}}$ and as a result $G \equiv_s^p G'$. Recall the construction of the p-kernel of G as $G^{p\mathcal{K}} = (A, \mathcal{R}^{p\mathcal{K}})$, where $\mathcal{R}^{p\mathcal{K}} = \mathcal{R} \setminus \{(a,b) \mid a \neq b, (a,a) \in \mathcal{R}, \{(b,a),(b,b)\} \cap \mathcal{R} \neq \emptyset\}$. We also established that

  $$I_{nst}(G) = \min\{|N| \mid \text{Extension}_s(\text{Induced}(G,N)) \neq \emptyset \text{ s.t. } N \subseteq \text{Nodes}(G)\}$$

  where for any subset $N \subseteq \mathcal{A}$, the induced subgraph Induced(G,N) is the induced subgraph with nodes $\mathcal{A} \setminus N$. In other words $I_{nst}(G)$ captures the minimum number of arguments to be removed from G to obtain a stable extension. By definition of the p-kernel construction, G and G' only differ in the removal of extra attacks, specifically (a,b) where a has a self-attack $(a,a) \in \mathcal{R}$ and where b is involved in attacks, $\{(b,a),(b,b)\} \cap R \neq \emptyset$. Since the differences between G and G' stem solely from attacks originating from self-attacking arguments, which cannot be part of a stable extension because they are not conflict-free, it holds that for any $N \subseteq \mathcal{A}$, Induced$(G,N)^{p\mathcal{K}}$ = Induced$(G',N)^{p\mathcal{K}}$. Therefore, Induced(G,N) has a stable extension iff Induced(G',N) does, meaning that the sets $N_G = \{N \subseteq \mathcal{A} \mid \text{Extension}_s(\text{Induced}(G,N)) \neq \emptyset\}$ and $N_{G'} = \{N \subseteq \mathcal{A} \mid \text{Extension}_s(\text{Induced}(G',N)) \neq \emptyset\}$ are identical. It follows that if $G \equiv_s^p G'$, then $I_{nst}(G) = \min\{|N| | N \in N_G\} = \min\{|N| | N \in N_{G'}\} = I_{nst}(G')$ holds.

- **(I_{n})** According to [30], conflict-freeness is preserved for two graphs G and G' where $G \equiv_s^p G'$. Recall that $I_{\text{n}}(G) = |\text{Extension}_n(G)| + |\text{selfAttacks}(G)| - 1$. Since Extension$_n(G)$ includes the maximal (w.r.t. $\subseteq$) conflict-free sets, and conflict-freeness is preserved, it holds that $|\text{Extension}_n(G)| = |\text{Extension}_n(G')|$. Additionally, self-attacks are not removed for the p-kernel construction s.t. $|\text{selfAttacks}(G)| = |\text{selfAttacks}(G')|$ by definition. As result, if $G \equiv_s^p G'$, then $I_{\text{n}}(G) = I_{\text{n}}(G')$.

- **(I_{p})** We have already established above that if $G \equiv_s^p G'$ then $I_{\text{pr}}(G) = I_{\text{pr}}(G') = |\text{Extension}_p(G)| - 1$. We can express I_{p} in terms of I_{pr}:

  $$I_{\text{p}} = |\text{Extension}_p(G)| + |\text{selfAttacks}(G)| - 1$$
  $$I_{\text{p}} = I_{\text{pr}}(G) + |\text{selfAttacks}(G)|.$$

  For $G^{p\mathcal{K}} = G'^{p\mathcal{K}}$ to hold, G and G' must have identical self-attacks s.t. under equality of $I_{\text{pr}}(G)$ and the number of self-attacks it follows that if $G \equiv_s^p G'$ then $I_{\text{p}}(G) = I_{\text{p}}(G')$.

- **(I_{dr})** Suppose that $G \equiv_s^p G'$ holds for two graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \mathcal{R}')$. By definition, Nodes$(G)$ = Nodes(G') and Arcs$(G) \neq$ Arcs(G'). Consider the following cases of attack relations:
  - $\mathcal{R} = \emptyset$ and $\mathcal{R}' = \emptyset$: Both graphs do not have any attacks s.t. $I_{\text{dr}}(G) = I_{\text{dr}}(G') = 0$.

– $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' \neq \emptyset$: Since $G \equiv_s^p G'$, $G^{p\mathcal{K}} = G'^{p\mathcal{K}}$ must hold, which means that except for the attacks redundant to the preferred extensions, G and G' both have at least one attack. As a result, $I_{\text{dr}}(G) = I_{\text{dr}}(G') = 1$.

– $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' = \emptyset$: To see that this case is contradictory analyze the graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \emptyset)$. Since G' contains no attacks, $G'^{p\mathcal{K}} = G' = (\mathcal{A}, \emptyset)$. However, because $\mathcal{R} \neq \emptyset$, G has at least one attack in its kernel $G^{p\mathcal{K}}$. Hence, under these attack conditions it cannot hold that $G^{p\mathcal{K}} \neq G'^{p\mathcal{K}}$ s.t. $G \not\equiv_s^p G'$ in first place.

The cases above show that if $G \equiv_s^p G'$ then $I_{\text{dr}}(G) = I_{\text{dr}}(G')$ holds.

Unless mentioned otherwise, for the following proofs consider argument graph L and its p-kernel $L^{p\mathcal{K}}$, as well as another graph L' which is the p-kernel of itself, which we label $L'^{p\mathcal{K}}$, all of which are displayed in Figure 7. Since $L^{p\mathcal{K}} = L'^{p\mathcal{K}}$, it holds that $L \equiv_s^p L'$. The semantics of L and L' are listed in Tables 17 and 3, respectively.

- **(I_{ngr})** $L \equiv_s^p L'$ but $I_{\text{ngr}}(L) = 2 \neq 0 = I_{\text{ngr}}(L')$.

$$\begin{aligned} I_{\text{ngr}}(L) &= |\text{Nodes}(L) \setminus (\text{Extension}_g(L) \cup \text{Attacked}(L))| \\ &= |\{a, b\} \setminus (\emptyset \cup \emptyset)| \\ &= |\{a, b\} \setminus \emptyset| = |\{a, b\}| = 2 \end{aligned}$$

$$\begin{aligned} I_{\text{ngr}}(L') &= |\text{Nodes}(L') \setminus (\text{Extension}_g(L') \cup \text{Attacked}(L'))| \\ &= |\{a, b\} \setminus (\{b\} \cup \{a\})| \\ &= |\{a, b\} \setminus \{a, b\}| = |\emptyset| = 0 \end{aligned}$$

- **(I_{win})** $L \equiv_s^p L'$ but $I_{\text{win}}(L) = \frac{1}{2} + 1 = \frac{3}{2} = 1.50 \neq 0.50 = \frac{1}{2} = I_{\text{win}}(L')$.
- **(I_{wout})** $L \equiv_s^p L'$ but $I_{\text{wout}}(L) = \frac{1}{2} + 1 = \frac{3}{2} = 1.50 \neq 2 = 1 + 1 = I_{\text{wout}}(L')$.
- **(I_{cc})** $L \equiv_s^p L'$ but $I_{\text{cc}}(L) = 2 \neq 1 = I_{\text{cc}}(L')$.
- **(I_{wcc})** $L \equiv_s^p L'$ but $I_{\text{wcc}}(L) = \frac{1}{2} + 1 = \frac{3}{2} = 1.50 \neq 1 = I_{\text{wcc}}(L')$.
- **(I_{wco})** For the following proof consider argument graph L_1 and its p-kernel $L_1^{p\mathcal{K}}$, as well as another graph L'_1 and its p-kernel, $L_1'^{p\mathcal{K}}$ in Figure 15. Since $L_1^{p\mathcal{K}} = L_1'^{p\mathcal{K}}$, it holds that $L_1 \equiv_s^p L'_1$. However, $I_{\text{wco}}(L_1) = (2-1)^2 = 1 \neq 0 = 0 + 0 = (1-1)^2 + (1-1)^2 = I_{\text{wco}}(L'_1)$.

Figure 15: Argument graphs L_1 (left) and $L'_1 = L_1^{p\mathcal{K}} = L_1'^{p\mathcal{K}}$ (right).

- **(I_{con})** $L \equiv_s^p L'$ but $I_{\text{con}}(L) = 3 \neq 2 = I_{\text{con}}(L')$.

- **(I_{ind})** For the following proof consider argument graph L_2 and its p-kernel $L_2^{p\mathcal{K}}$, as well as another graph L_2' and its p-kernel, $L_2'^{p\mathcal{K}}$ in Figure 16. Since $L_2^{p\mathcal{K}} = L_2'^{p\mathcal{K}}$, it holds that $L_2 \equiv_s^p L_2'$. Yet, $I_{\text{ind}}(L_2) = 3 \neq 2 = I_{\text{ind}}(L_2')$.

Figure 16: Argument graphs L_2 (left) and $L_2' = L_2^{p\mathcal{K}} = L_2'^{p\mathcal{K}}$ (right).

- **(I_{db})** It holds that $L \equiv_s^p L'$. However, the global distances for the graphs are different with $GD(L) = 1 + 1 + 1 + 2 = 5$ and $GD(L') = 1 + 3 + 1 + 3 = 8$. Since both graphs have the same number of nodes it holds for both that $\max = 2^2 \times (2 + 1) = 12$ and $\min = 2^2 = 4$.

$$I_{db}(G) = \frac{\max - GD(G)}{\max - \min}$$

$$I_{db}(L) = \frac{12 - 5}{12 - 4} = \frac{7}{8} \approx 0.88 \neq 0.50 = \frac{1}{2} = \frac{12 - 8}{12 - 4} = I_{db}(L').$$

□

Proposition 4.8 *g-Exchange* is satisfied by I_{ngr}, I_{n}, I_{dr}, and I_{ind} but not by I_{pr}, I_{p}, I_{nst}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{wco}, I_{con}, and I_{db}.

Proof.

- **(I_{ngr})** Recall that I_{ngr}, counts the arguments not in the grounded extension and not attacked by a member of the grounded extension:

$$I_{\text{ngr}}(G) = |\text{Nodes}(G) \setminus (\text{Extension}_g(\text{G}) \cup \text{Attacked}(G))|$$

where $\text{Attacked}(G) = \{b \mid (a, b) \in \text{Arcs}(G) \text{ and } a \in \text{Extension}_g(\text{G})\}$. Consider two graphs G and G' with $G \equiv_s^g G'$, requiring $G^{g\mathcal{K}} = G'^{g\mathcal{K}}$, s.t. $\text{Extension}_g(\text{G}) = \text{Extension}_g(\text{G}')$ and $\text{Nodes(G)} = \text{Nodes(G}')$. Since $\text{Attacked}(G)$ depends on the grounded extension, $\text{Extension}_g(\text{G})$, and the attack relations, we check whether the g-kernel construction changes the set of $\text{Attacked}(G)$ or not. As established in Section 2.4, the g-kernel is constructed as $G^{g\mathcal{K}} = (A, \mathcal{R}^{g\mathcal{K}})$, where $\mathcal{R}^{g\mathcal{K}} = \mathcal{R} \setminus \{(a, b) \mid a \neq b, (b, b) \in \mathcal{R}, \{(a, a), (b, a)\} \cap \mathcal{R} \neq \emptyset\}$. In other words, we remove (a, b) if

  - a and b are self-attacking (case 1), or
  - b is self-attacking, and b also attacks a (case 2), or
  - a and b are self-attacking, and b also attacks a (case 3).

The first case is equivalent to two self-attacking arguments with no arc between them, because self-attacking arguments $a, b \notin$ Extension$_g$(G). Based on the definiton of Attacked(G) above, the arc (a, b) does not contribute to Attacked(G). For the second case, it holds that $a \notin$ Extension$_g$(G) as it is attacked and $b \notin$ Extension$_g$(G) because it is self-attacking. It follows that (a, b) does not contribute to Attacked(G) as well. Similarly, for the combination of cases 1 and 2, illustrated in case 3 it holds that $a, b \notin$ Extension$_g$(G) and therefore the arc (a, b) does not contribute to Attacked(G). Hence, Attacked(G) = Attacked($G^{g\mathcal{K}}$) holds. Given that $G^{g\mathcal{K}} = G'^{g\mathcal{K}}$ it follows that Attacked(G) = Attacked($G^{g\mathcal{K}}$) = Attacked($G'^{g\mathcal{K}}$) = Attacked(G'). We have shown that the g-kernel construction, preserves the Nodes(G), Extension$_g$(G), selfAttacks(G) and Attacked(G). Therefore, it holds that $I_{\text{ngr}}(G) = I_{\text{ngr}}(G')$ if $G \equiv_s^g G'$.

- **($I_{\mathbf{n}}$)** According to [30], conflict-freeness is preserved for two graphs G and G' where $G \equiv_s^g G'$. Recall that $I_{\text{n}}(G) = |\text{Extension}_n(G)| + |\text{selfAttacks}(G)| - 1$. Since Extension$_n(G)$ includes the maximal (w.r.t. $\subseteq$) conflict-free sets, and conflict-freeness is preserved, it holds that Extension$_n(G)$ = Extension$_n(G')$. Additionally, self-attacks are not removed for the g-kernel construction meaning that for $G^{g\mathcal{K}} = G'^{g\mathcal{K}}$ to hold, selfAttacks(G) = selfAttacks(G') must also hold. As result, if $G \equiv_s^g G'$ then $I_{\text{n}}(G) = I_{\text{n}}(G')$.

- **($I_{\mathbf{dr}}$)** Suppose that $G \equiv_s^g G'$ holds for two graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \mathcal{R}')$. By definition, Nodes($G$) = Nodes($G'$) and Arcs($G$) $\neq$ Arcs(G'). We distinguish between the following attack cases:
  - $\mathcal{R} = \emptyset$ and $\mathcal{R}' = \emptyset$: Since both graphs have no attacks, it follows that $I_{\text{dr}}(G) = I_{\text{dr}}(G') = 0$.
  - $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' \neq \emptyset$: G and G' both have at least one attack by definition s.t. $I_{\text{dr}}(G) = I_{\text{dr}}(G') = 1$.
  - $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' = \emptyset$: To see that this case is contradictory analyze the graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \emptyset)$. Since G' contains no attacks, $G'^{g\mathcal{K}} = G' = (\mathcal{A}, \emptyset)$. However, because $\mathcal{R} \neq \emptyset$, G has at least one attack in its kernel $G^{g\mathcal{K}}$. It holds that $G^{g\mathcal{K}} \neq G'^{g\mathcal{K}}$ s.t. $G \equiv_s^g G'$ cannot hold.

  The cases show that if $G \equiv_s^g$, it also holds that $I_{\text{dr}}(G) = I_{\text{dr}}(G')$.

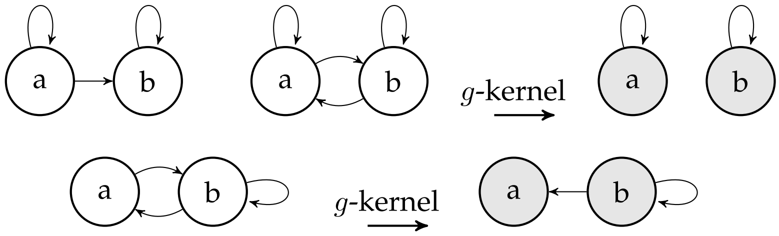


Figure 17: Argument graphs L_3 (upper left), L_4 (upper middle), and their g-kernel $L_{3,4}^{g\mathcal{K}} = L_3^{g\mathcal{K}} = L_4^{g\mathcal{K}}$ (upper right) as well as L_5 (lower left) and its g-kernel $L_5^{g\mathcal{K}}$ (lower right).

- **(I_{ind})** Recall how for a graph G the g-kernel is constructed: $G^{g\mathcal{K}} = (A, \mathcal{R}^{g\mathcal{K}})$, where $\mathcal{R}^{g\mathcal{K}} = \mathcal{R} \setminus \{(a,b) \mid a \neq b, (b,b) \in \mathcal{R}, \{(a,a),(b,a)\} \cap \mathcal{R} \neq \emptyset\}$. We iterate through all the attack removal cases:
  - $\mathcal{R} = \emptyset$: $I_{\text{ind}}(G) = I_{\text{ind}}(G') = 0$
  - $\mathcal{R} \neq \emptyset$: we remove (a,b) if
    - a and b are self-attacking, or
    - a and b are self-attacking, and b also attacks a, or
    - b is self-attacking, and b also attacks a.

  It is visible that for all cases, the arguments a and b remain attacked before and after the removal of redundant attacks. We have illustrated this in Figure 17, where the kernels of the graphs are in the same line of the graphs, shaded in gray. This showcases that the construction of the g-kernel preserves the number of attacked arguments. Since $G \equiv_s^g G'$, $G^{g\mathcal{K}} = G'^{g\mathcal{K}}$ must hold, which in turn also means that all nodes that have at least one attack in G also have at least one attack in G'. Since $I_{\text{ind}}(G)$ quantifies the number of attacked arguments, it follows that $I_{\text{ind}}(G) = I_{\text{ind}}(G')$ also holds.

Figure 18: Argument graphs L_6 (left) and $L_6' = L_6^{g\mathcal{K}} = L_6'^{g\mathcal{K}}$ (right).

- **(I_{pr} and I_{p})** Consider the following counterexample with argument graph L_6 and another graph L_6' in Figure 18, as well as their semantics in Tables 18 and 19, respectively. Their g-kernels $L_6^{g\mathcal{K}} = L_6'^{g\mathcal{K}}$ s.t. $L_6 \equiv_s^g L_6'$ holds are in the same Figure. But $I_{\text{pr}}(L_6) = 2 - 1 = 1 \neq 0 = 1 - 1 = I_{\text{pr}}(L_6')$. Similarly, $I_{\text{p}}(L_6) = 2 + 1 - 1 = 2 \neq 1 = 1 + 1 - 1 = I_{\text{p}}(L_6')$.

Set(L_6)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
$\emptyset$	✓	✓	✓	×	✓	×	×
{a}	✓	✓	✓	✓	×	✓	✓
{b}	✓	✓	✓	✓	×	×	✓
{c}	×	×	×	×	×	×	×

Table 18: Semantics of argument graph L_6.

Set(L'_6)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
$\emptyset$	✓	✓	✓	×	✓	×	×
{a}	✓	×	×	×	×	×	✓
{b}	✓	✓	✓	✓	×	×	✓
{c}	×	×	×	×	×	×	×

Table 19: Semantics of argument graph L'_6.

- **(I_{nst})** Consider graph L_5 and its g-Kernel $L_5^{g\mathcal{K}}$ in Figure 17. Observe another graph P in Figure 9 which is the g-kernel of itself, which we label $P^{g\mathcal{K}}$. $L_5^{g\mathcal{K}} = P^{g\mathcal{K}}$ holds. We have calculated in Example 3.7 that $I_{\text{nst}}(P) = 1$. We calculate $I_{\text{nst}}(L_5) = 0$ based on the semantics of L_5 in Table 20.

$$N = \emptyset : \text{Extension}_s(\text{Induced}(L_5, \emptyset)) = \text{Extension}_s(\{a, b\}) = \{a\}$$

$L_5 \equiv_s^g P$, but $I_{\text{nst}}(L_5) = 0 \neq 1 = I_{\text{nst}}(P)$.

Set(L_5)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
$\emptyset$	✓	✓	✓	×	✓	×	×
{a}	✓	✓	✓	✓	×	✓	✓
{b}	×	×	×	×	×	×	×

Table 20: Semantics of argument graph L_5.

For the following proofs, unless mentioned otherwise, consider argument graph L in Figure 7 and its g-kernel $L^{g\mathcal{K}}$, as well as another graph L'' and which its own g-kernel which we label $L''^{g\mathcal{K}}$. Since $L^{g\mathcal{K}} = L''^{g\mathcal{K}}$, it holds that $L \equiv_s^g L''$.

- **(I_{win})** $L \equiv_s^g L''$ but $I_{\text{win}}(L) = 1 + \frac{1}{2} = \frac{3}{2} = 1.50 \neq 2 = 1 + 1 = I_{\text{win}}(L'')$.
- **(I_{wout})** $L \equiv_s^g L''$ but $I_{\text{wout}}(L) = 1 + \frac{1}{2} = \frac{3}{2} = 1.50 \neq 0.50 = \frac{1}{2} = I_{\text{wout}}(L'')$.
- **(I_{cc})** $L \equiv_s^g L''$ but $I_{\text{cc}}(L) = 2 \neq 1 = I_{\text{cc}}(L'')$.

- **(I_{wcc})** $L \equiv_s^g L''$ but $I_{wcc}(L) = 1 + \frac{1}{2} = \frac{3}{2} = 1.50 \neq 1 = I_{wcc}(L'')$.

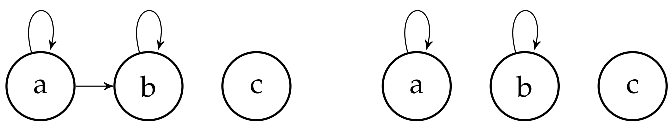


Figure 19: Argument graphs L_7 (left) and $L'_7 = L_7^{g\mathcal{K}} = L_7'^{g\mathcal{K}}$ (right).

- **(I_{wco})** Consider the graph L_7 and its g-kernel $L_7^{g\mathcal{K}}$, and L'_7 and its g-kernel $L_7'^{g\mathcal{K}}$ in Figure 19. $L_7 \equiv_s^g L'_7$ but $I_{wco}(L_7) = (2-1)^2 + (1-1)^2 = 1 \neq 0 = (1-1)^2 + (1-1)^2 + (1-1)^2 = I_{wco}(L'_7)$.
- **(I_{con})** $L \equiv_s^g L''$ but $I_{con}(L) = 3 \neq 2 = I_{con}(L'')$.
- **(I_{db})** $L \equiv_s^g L''$ and as calculated in the proof for I_{db} under Proposition 4.7, $I_{db}(L) = \frac{7}{8} \approx 0.88$. As for L'', $GD(L'') = 1+3+1+3 = 8$, $\max = 2^2 \times (2+1) = 12$, and $\min = 2^2 = 4$.

$$I_{db}(G) = \frac{\max - GD(G)}{\max - \min}$$

$$I_{db}(L) = \frac{7}{8} \approx 0.88 \neq 0.50 = \frac{1}{2} = \frac{12-8}{12-4} = I_{db}(L'').$$

□

Proposition 4.9 *s-Exchange* is satisfied by I_{nst}, I_n and I_{dr}. The measures I_{pr}, I_p, I_{ngr}, I_{dr}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{wco}, I_{con}, I_{ind}, and I_{db} violate s-Exchange.

Proof.

- **(I_{nst})** For graphs G and G' it holds that $G^{s\mathcal{K}} = G'^{s\mathcal{K}}$ s.t. $G \equiv_s^s G'$. Recall the s-kernel construction: $G^{s\mathcal{K}} = (\mathcal{A}, \mathcal{R}^{s\mathcal{K}})$, where $\mathcal{R}^{s\mathcal{K}} = \mathcal{R} \setminus \{(a,b) \mid a \neq b, (a,a) \in \mathcal{R}\}$. Also recall that I_{nst} is the smallest number of nodes $N \subseteq \mathcal{A}$ that must be removed so that the induced subgraph Induced(G, N) has at least one stable extension, if the graph does not have one already. Since G and G' have the same s-kernel it also holds that for every subset N of nodes, the induced subgraphs Induced(G, N) and Induced(G', N) have the same s-kernel. Since the existence of a stable extension in an argument graph depends on its s-kernel, it follows that Extension$_s$(Induced$(G, N)) \neq \emptyset$ iff Extension$_s$(Induced$(G', N)) \neq \emptyset$. Now assume that $I_{nst}(G) = n$ where $n \in \mathbb{Z}, \quad n \geq 1$, meaning there exists a subset $N \subseteq \mathcal{A}$ with $|N| = n$ s.t. Induced(G, N) has a stable extension. Since $G \equiv_s^s G'$, Induced$(G, N)^{s\mathcal{K}} =$ Induced$(G', N)^{s\mathcal{K}}$, and thus Induced(G, N) and Induced(G', N) have identical stable extensions. This implies that $I_{nst}(G')$ also has a stable extension, and therefore $I_{nst}(G') \leq n$ must hold. Now suppose $I_{nst}(G') = m$ where $m \in \mathbb{Z}, \quad m \geq 1$. Then it must also hold that $I_{nst}(G) \leq m$.

Combining these assumptions results in $I_{\text{nst}}(G) \leq I_{\text{nst}}(G') \leq n = I_{\text{nst}}(G)$ and it follows that $I_{\text{nst}}(G') = n$. Similarly, if we start with $I_{\text{nst}}(G') = m$, the same applies: $I_{\text{nst}}(G') \leq I_{\text{nst}}(G) \leq m = I_{\text{nst}}(G')$, resulting in $I_{\text{nst}}(G) = m$. $I_{\text{nst}}(G) = I_{\text{nst}}(G')$ holds if $G \equiv^s_s G'$.

- **(I_{n})** According to [30], conflict-freeness is preserved for two graphs G and G' where $G \equiv^s_s G'$. Recall that $I_{\text{n}}(G) = |\text{Extension}_n(G)| + |\text{selfAttacks}(G)| - 1$. Since $\text{Extension}_n(G)$ includes the maximal (w.r.t. $\subseteq$) conflict-free sets, and conflict-freeness is preserved, it holds that $\text{Extension}_n(G) = \text{Extension}_n(G')$. Additionally, self-attacks are not removed for the s-kernel construction meaning that for $G^{s\mathcal{K}} = G'^{s\mathcal{K}}$ to hold, selfAttacks(G) = selfAttacks(G') must hold. As result, if $G \equiv^s_s G'$ then $I_{\text{n}}(G) = I_{\text{n}}(G')$.

- **(I_{dr})** Suppose that $G \equiv^s_s G'$ holds for two graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \mathcal{R}')$. By definition, Nodes($G$) = Nodes($G'$) and Arcs($G$) $\neq$ Arcs(G'). We distinguish between the following attack cases:
  - $\mathcal{R} = \emptyset$ and $\mathcal{R}' = \emptyset$: $I_{\text{dr}}(G) = I_{\text{dr}}(G') = 0$, by definition.
  - $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' \neq \emptyset$: G and G' both have at least one attack s.t. $I_{\text{dr}}(G) = I_{\text{dr}}(G') = 1$.
  - $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' = \emptyset$: This case is contradictory. Analyze the graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \emptyset)$. Since G' contains no attacks, $G'^{s\mathcal{K}} = G' = (\mathcal{A}, \emptyset)$. However, because $\mathcal{R} \neq \emptyset$, G has at least one attack in its kernel $G^{s\mathcal{K}}$. It holds that $G^{s\mathcal{K}} \neq G'^{s\mathcal{K}}$ s.t. $G \equiv^s_s G'$ cannot hold.

  The valid cases show that if $G \equiv^s_s$, it also holds that $I_{\text{dr}}(G) = I_{\text{dr}}(G')$.

Figure 20: Argument graphs L_8 (left) and $L'_8 = L_8^{s\mathcal{K}} = L_8'^{s\mathcal{K}}$ (right).

- **(I_{pr} and I_{p})** Observe argument graphs L_8 and L'_8, their semantics in Table 21 and 22 respectively, and their s-kernels $L_8^{s\mathcal{K}} = L_8'^{s\mathcal{K}}$ in Figure 20 s.t. $L_8 \equiv^s_s L'_8$ holds. But $I_{\text{pr}}(L_8) = 1 - 1 = 0 \neq 1 = 2 - 1 = I_{\text{pr}}(L'_8)$. Similarly, $I_{\text{p}}(L_8) = 1 + 1 - 1 = 1 \neq 2 = 2 + 1 - 1 = I_{\text{p}}(L'_8)$.

Set(L_8)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	✓	×	✓	×	×
{a}	✓	×	×	×	×	×	✓
{b}	✓	✓	✓	✓	×	×	✓
{c}	×	×	×	×	×	×	×

Table 21: Semantics of argument graph L_8.

Set(L'_8)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
$\emptyset$	✓	✓	✓	✗	✓	✗	✗
{a}	✓	✓	✓	✓	✗	✗	✓
{b}	✓	✓	✓	✓	✗	✗	✓
{c}	✗	✗	✗	✗	✗	✗	✗

Table 22: Semantics of argument graph L'_8.

For the following proofs, unless mentioned otherwise, consider argument graph L in Figure 7 and its s-kernel $L^{s\mathcal{K}}$, as well as another graph L' in the same Figure, which is the s-kernel of itself, and which we label $L'^{s\mathcal{K}}$. Their semantics are visible in Table 17 and 3, respectively. Since $L^{s\mathcal{K}} = L'^{s\mathcal{K}}$, it holds that $L \equiv_s^s L'$. The computations for the following proofs are equivalent to those in the corresponding proofs under Proposition 4.7.

- **(I_{ngr})** $L \equiv_s^s L'$ but $I_{\text{ngr}}(L) = 2 \neq 0 = I_{\text{ngr}}(L')$.
- **(I_{win})** $L \equiv_s^s L'$ but $I_{\text{win}}(L) = 1.50 \neq 0.50 = I_{\text{win}}(L')$.
- **(I_{wout})** $L \equiv_s^s L'$ but $I_{\text{wout}}(L) = 1.50 \neq 2 = I_{\text{wout}}(L')$.
- **(I_{cc})** $L \equiv_s^s L'$ but $I_{\text{cc}}(L) = 2 \neq 1 = I_{\text{cc}}(L')$.
- **(I_{wwc})** $L \equiv_s^s L'$ but $I_{\text{wcc}}(L) = 1.50 \neq 1 = I_{\text{wcc}}(L')$.

Figure 21: Argument graphs L_9 (left) and $L'_9 = L_9^{s\mathcal{K}} = L_9'^{s\mathcal{K}}$ (right).

- **(I_{wco})** For the following proof consider argument graph L_9 in Figure 21 and its s-kernel $L_9^{s\mathcal{K}}$, as well as another graph L'_9 and its s-kernel, $L_9'^{s\mathcal{K}}$. Since $L_9^{s\mathcal{K}} = L_9'^{s\mathcal{K}}$, it holds that $L_9 \equiv_s^s L'_9$. However, $I_{\text{wco}}(L_9) = (3-1)^2 = 4 \neq 1 = (1-1)^2 + (2-1)^2 = I_{\text{wco}}(L'_9)$.
- **(I_{con})** $L \equiv_s^s L'$ but $I_{\text{con}}(L) = 3 \neq 2 = I_{\text{con}}(L')$.
- **(I_{ind})** Recall graphs L_9 and L'_9 from the counterexample for I_{wco}. $L_9 \equiv_s^s L'_9$ but $I_{\text{ind}}(L_9) = 3 \neq 2 = I_{\text{wco}}(L'_9)$.
- **(I_{db})** $L \equiv_s^s L'$, but $I_{\text{db}}(L) = \frac{7}{8} \approx 0.88 \neq 0.50 = I_{\text{db}}(L')$ □

Proposition 4.10 *naive-Exchange* is satisfied by I_{n}, I_{dr} and I_{wco}. It is not satisfied by I_{pr}, I_{p}, I_{ngr}, I_{nst}, I_{win}, I_{wout}, I_{cc}, I_{wcc}, I_{con}, I_{ind}, and I_{db}.

Proof.

- **(I_{n})** Recall that $I_{\mathrm{n}} = |\text{Extension}_n(G)| + |\text{selfAttacks}(G)| - 1$ and that naive extensions are maximal (w.r.t. $\subseteq$) conflict-free extensions. For two graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \mathcal{R}')$, $G \equiv_s^{naive} G'$ holds, if the graphs have the same arguments and naive extensions. Hence, by definition for two graphs $G \equiv_s^{naive} G'$, it holds that $\text{Extension}_n(G) = \text{Extension}_n(G')$ and $\text{Nodes}(G) = \text{Nodes}(G')$. We consider the self-attacks. Suppose for contradiction that there exists an argument $a \in \mathcal{A}$ s.t. $(a, a) \in \mathcal{R}$ but $(a, a) \notin \mathcal{R}'$. In G, $\{a\}$ is not conflict-free due to the self-attack and in G', $\{a\}$ is conflict-free because there is no self-attack. If $\{a\}$ is a naive extension in G', it cannot be one in G, contradicting the equality of naive extensions. If $\{a\}$ is not a naive extension in G', there must exist a larger conflict-free set $\mathcal{S}$ that includes $\{a\}$ in G'. However, in G, any set containing a is not conflict-free. This leads to a contradiction once more. Therefore, it must hold that $(a, a) \in \mathcal{R} \Leftrightarrow (a, a) \in \mathcal{R}'$ and that G and G' must have the same self-attacks. Consequently, it holds that if $G \equiv_s^{naive} G'$ then $I_{\mathrm{n}}(G) = I_{\mathrm{n}}(G')$.

- **(I_{dr})** Suppose that for $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \mathcal{R}')$, $\text{Nodes}(G) = \text{Nodes}(G')$ and $\text{Extension}_n(G) = \text{Extension}_n(G)$ s.t. $G \equiv_s^{naive} G'$ holds. We distinguish between the following attack cases:
  - $\mathcal{R} = \emptyset$ and $\mathcal{R}' = \emptyset$: By definition $I_{\mathrm{dr}}(G) = I_{\mathrm{dr}}(G') = 0$.
  - $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' \neq \emptyset$: By definition $I_{\mathrm{dr}}(G) = I_{\mathrm{dr}}(G') = 1$.
  - $\mathcal{R} \neq \emptyset$ and $\mathcal{R}' = \emptyset$: To see that this case is contradictory analyze the graphs $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}, \emptyset)$. Since G' has empty attack relations, every subset $\mathcal{A}$ is conflict-free, resulting in the naive extension $\{\mathcal{A}\}$. As $\mathcal{R} \neq \emptyset$ in G, there exists at least one attack $(a, b) \in \mathcal{R}$, where $a, b \in \mathcal{A}$ s.t. $\{a, b\}$ is not conflict-free. Therefore, $\{\mathcal{A}\}$ cannot be a naive extension in G. This in turn means that G and G' do not have the same naive extensions. It follows that $G \not\equiv_s^{naive} G'$ holds.

  The valid cases show that if $G \equiv_s^{naive}$, it also holds that $I_{\mathrm{dr}}(G) = I_{\mathrm{dr}}(G')$.

- **(I_{wco})** Assume that $G \equiv_s^{naive} G'$ holds for two graphs G and G'. Strong equivalence demands that both graphs have the same arguments and same maximal (w.r.t. $\subseteq$) conflict-free sets. For each component, I_{wco} sums up $(n-1)^2$ where $n \geq 1$ is the number of nodes. Since the number of nodes n is constant under strong equivalence, we check whether for G and G' the number of components as well as their size remain the same or not under $G \equiv_s^{naive} G'$. Let the components of G partition $\mathcal{A}$ into disjoint subsets $C_1, C_2, \ldots, C_k$ where $k \in \mathbb{Z}$, $k > 0$. Clearly any G is the union of its disjoint components C_i for $i \geq 1$. Since these components do not attack each other, the naive extension for the entire graph is $\text{Extension}_n(G) = \{\mathcal{S}_1 \cup \mathcal{S}_2 \cup ... \cup \mathcal{S}_k | \mathcal{S}_i \in \text{Extension}_n(G(C_i))\}$, reflecting that each component C_i contributes independently to naive extensions. Now consider the following contradictions:

- Assume that G has more components than G'. Then, the naive extensions of G would be would be formed from different component choices than those of G'. This would result in $\text{Extension}_n(G) \neq \text{Extension}_n(G')$, contradicting our assumption. Therefore, G and G' must have the same number of components.
- Suppose, that one component in G contains more or fewer arguments than the corresponding component in G'. Then, the set of maximally conflict-free subsets would be different in that component. This would lead to $\text{Extension}_n(G) \neq \text{Extension}_n(G')$, contradicting our assumption once more. Thus, each corresponding component in G and G' must have the same number of arguments.

Since the number of nodes n, as well as the number of components and their sizes are equal to each other for the graphs G and G', under the condition $G \equiv_s^{naive} G'$, and I_{wco} sums up $(n-1)^2$ where $n \geq 1$ is the number of nodes, for each component, it holds that $I_{\text{wco}}(G) = I_{\text{wco}}(G')$.

For the following proofs, unless mentioned otherwise, consider the following argument graphs $M+A$ consisting of graph M in Figure 8 and A in Figure 1 and graph $M+C'$ consisting of components M, and C' in Figure 3. Both graphs have the same naive extensions $\{a,x\}$ and $\{b,x\}$ and the same arguments s.t. $M+A \equiv_s^{naive} M+C'$. The semantics of $M+A$ are available in Table 11 and the semantics of graph $M+C'$ are apparent in Table 23.

Set(M+C')	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
∅	✓	✓	×	×	×	×	×
{a}	✓	×	×	×	×	×	×
{b}	✓	✓	×	×	×	×	×
{x}	✓	✓	✓	×	✓	×	×
{a, x}	✓	×	×	×	×	×	✓
{b, x}	✓	✓	✓	✓	×	✓	✓

Table 23: Semantics of argument graph $M+C'$.

- **(I_{pr} and I_{p})** $M+A \equiv_s^{naive} M+U$ but $I_{\text{pr}}(M+A) = 2-1 = 1 \neq 0 = 1-1 = I_{\text{pr}}(M+C')$, and $I_{\text{p}}(M+A) = 2+0-1 = 1 \neq 0 = 1+0-1 = I_{\text{p}}(M+C')$.
- **(I_{ngr})** For the graphs L and L' in Figure 7, $L \equiv_s^{naive} L'$ because the graphs have the same arguments and naive extensions as visible in Tables 17 and 3,

respectively. But $I_{\text{ngr}}(L) = 2 \neq 0 = I_{\text{ngr}}(L')$ as calculated in the the proof for I_{ngr} under Proposition 4.7.

- **(I_{nst})** For the graphs L_5 in Figure 17 and P in Figure 9, $L_5 \equiv_s^{naive} P$ since the graphs have the same arguments and naive extensions apparent in Table 13 and 20. Yet, $I_{\text{nst}}(L_5) = 0 \neq 1 = I_{\text{nst}}(P)$ as calculated in the proof for I_{nst} under Proposition 4.8.
- **(I_{win})** $M + A \equiv_s^{naive} M + C'$ but $I_{\text{win}}(M + A) = 1 + 1 = 2 \neq 1 = I_{\text{win}}(M + C')$.
- **(I_{wout})** $M + A \equiv_s^{naive} M + C'$ but $I_{\text{wout}}(M + A) = 1 + 1 = 2 \neq 1 = I_{\text{wout}}(M + C')$.
- **(I_{cc})** $M + A \equiv_s^{naive} M + C'$ but $I_{\text{cc}}(M + A) = 1 \neq 0 = I_{\text{cc}}(M + C')$.
- **(I_{wcc})** $M + A \equiv_s^{naive} M + C'$ but $I_{\text{wcc}}(M + A) = \frac{1}{2} = 0.50 \neq 0 = I_{\text{wcc}}(M + C')$.
- **(I_{con})** $M + A \equiv_s^{naive} M + C'$ but $I_{\text{con}}(M + A) = 2 \neq 1 = I_{\text{con}}(M + C')$.
- **(I_{ind})** $M + A \equiv_s^{naive} M + C'$ but $I_{\text{ind}}(M + A) = 2 \neq 1 = I_{\text{ind}}(M + C')$.
- **(I_{db})** $M + A \equiv_s^{naive} M + C'$. We compute the global distances fpr the graphs s.t. $GD(M + A) = (3 \times 4) + (2 \times (1 + 2 + 4)) = 26$ and $GD(M + C') = (1 + 4 + 4) + (2 \times (3 \times 4)) = 33$. Since both graphs have the same quantity of nodes, $\max = 3^2 \times (3 + 1) = 9 \times 4 = 36$ and $\min = 3^2 = 9$.

$$I_{\text{db}}(G) = \frac{\max - GD(G)}{\max - \min}$$

$$I_{\text{db}}(M + A) = \frac{36 - 26}{36 - 9} = \frac{10}{27} \approx 0.37 \neq 0.11 \approx \frac{1}{9} = \frac{36 - 33}{36 - 9} = I_{\text{db}}(M + C')$$

□

4.2.4 Additivity Postulate Propositions and Proofs

Proposition 4.11 *Disjoint Additivity* is satisfied by I_{con} and I_{ind} but not by I_{n}, I_{p}, and I_{db}.

Proof.

- **(I_{con})** Consider two arbitrary graphs $G_1 = (\mathcal{A}_1, \mathcal{R}_1)$ and $G_2 = (\mathcal{A}_2, \mathcal{R}_2)$ that are disjoint. Suppose G_1 has n arcs and G_2 has m arcs where for both $n, m \in \mathbb{Z}, n, m \geq 0$ s.t. $I_{\text{con}}(G_1) = |\mathcal{R}_1| = n$ and $I_{\text{con}}(G_2) = |\mathcal{R}_2| = m$. Since $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$, their union, $G_1 + G_2 = (\mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2)$, also does not share any arcs across G_1 and G_2 s.t. $\mathcal{R}_1 \cap (\mathcal{A}_1 \times \mathcal{A}_2) = \emptyset$ and $\mathcal{R}_2 \cap (\mathcal{A}_2 \times \mathcal{A}_1) = \emptyset$ hold. Given that no new arcs are added between G_1 and G_2 when the graph is joint, $I_{\text{con}}(G_1 + G_2) = |\mathcal{R}_1 \cup \mathcal{R}_2| = |\mathcal{R}_1| + |\mathcal{R}_2| = n + m$. It automatically follows that $I_{\text{con}}(G_1 + G_2) = I_{\text{con}}(G_1) + I_{\text{con}}(G_2)$.

- **(I_{ind})** Once again consider two arbitrary graphs $G_1 = (\mathcal{A}_1, \mathcal{R}_1)$ and $G_2 = (\mathcal{A}_2, \mathcal{R}_2)$ that are disjoint. Suppose G_1 has n attacked nodes and G_2 has m attacked nodes where for both $n, m \in \mathbb{Z}$, $n, m \geq 0$. $I_{\text{ind}}(G_1) = n$ and $I_{\text{ind}}(G_2) = m$. Since $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$, their union, $G_1 + G_2 = (\mathcal{A}_1 \cup \mathcal{A}_2, \mathcal{R}_1 \cup \mathcal{R}_2)$, also does not share any arcs across G_1 and G_2 s.t. $\mathcal{R}_1 \cap (\mathcal{A}_1 \times \mathcal{A}_2) = \emptyset$ and $\mathcal{R}_2 \cap (\mathcal{A}_2 \times \mathcal{A}_1) = \emptyset$ hold. Given that no new attacks are added between G_1 and G_2, the number of attacked arguments is simply the summation of the quantity of attacked arguments in G_1 and G_2 s.t. $I_{\text{ind}}(G_1 + G_2) = n + m$. It holds that $I_{\text{ind}}(G_1 + G_2) = I_{\text{ind}}(G_1) + I_{\text{ind}}(G_2)$.

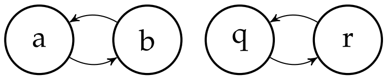


Figure 22: Argument graph $A + V$.

- **(I_{n})** Observe graph A in Figure 1, whose semantics are available in Table 10. $I_{\text{n}}(A) = 2 + 0 - 1 = 1$. Now visualize graph A being joint with disjoint (and isomorphic) graph V in Figure 14, for which $I_{\text{n}}(V) = 2 + 0 - 1 = 1$. The joint graph $A + V$ is visible in Figure 22. $I_{\text{n}}(A + V) = 4 + 0 - 1 = 3$ based on the semantics in Table 24. We show that I_{n} does not comply with Disjoint Additivity as $I_{\text{n}}(A) + I_{\text{n}}(V) = 1 + 1 = 2 \neq 3 = I_{\text{n}}(A + V)$.

Set($A + V$)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
∅	✓	✓	✓	×	✓	×	×
{a}	✓	✓	✓	×	×	×	×
{b}	✓	✓	✓	×	×	×	×
{q}	✓	✓	✓	×	×	×	×
{r}	✓	✓	✓	×	×	×	×
{a, q}	✓	✓	✓	✓	×	✓	✓
{a, r}	✓	✓	✓	✓	×	✓	✓
{b, q}	✓	✓	✓	✓	×	✓	✓
{b, r}	✓	✓	✓	✓	×	✓	✓

Table 24: Semantics of argument graph $A + V$.

- **(I_{p})** Once again examine graph A in Figure 1 and another graph V, visible in Figure 14 as counterexamples. Since both graphs are isomorphic to each other, $I_{\text{p}}(A) = I_{\text{p}}(V) = 2 + 0 - 1 = 1$ according to the semantics in Table 10. Now observe the joint graph $A + V$ in Figure22, consisting of the disjoint components. Based on Table 24, $I_{\text{p}}(A+V) = 4+0-1 = 3$. But $I_{\text{n}}(A)+I_{\text{n}}(V) = 1 + 1 = 2 \neq 3 = I_{\text{n}}(A + V)$.

- **(I_{db})** Take into account the following counterexample once more involving graphs A and V in Figure 1 and 14, respectively. Given that both graphs are isomorphic to each other, $I_{\text{db}}(A) = I_{\text{db}}(V) = 0.75$ as computed in Example 3.28. Observe their joint graph $A + V$ in Figure 22. $GD(A + V) = 4 \times (1 + 2 + 5 + 5) = 52$. As $n = 4$, $\max = 4^2 \times (4 + 1) = 80$ and $\min = 4^2 = 16$.

$$I_{\text{db}}(A + V) = \frac{\max - GD(A + V)}{\max - \min} = \frac{80 - 52}{80 - 16} = \frac{7}{16} \approx 0.44$$

$$I_{\text{db}}(A) + I_{\text{db}}(V) = 0.75 + 0.75 = 1.50\text{[4]} \neq 0.44 \approx I_{\text{db}}(A + V)$$

□

Proposition 4.12 *Super Addivity* is not satisfied by I_{n}, I_{p}, I_{con}, I_{ind}, and I_{db}.

Proof.

- **(I_{n})** Consider argument graphs W, Y and their joint graph $W + Y$ in Figure 23 and their semantics available in Table 25, 26, and 27, respectively, as counterexamples. $I_{\text{n}}(W) = 1 + 1 - 1 = 1$, $I_{\text{n}}(Y) = 1 + 1 - 1 = 1$, and $I_{\text{n}}(W + Y) = 1 + 1 - 1 = 1$ s.t. $I_{\text{n}}(W + Y) = 1 \not\geq 2 = 1 + 1 = I_{\text{n}}(W) + I_{\text{n}}(Y)$.

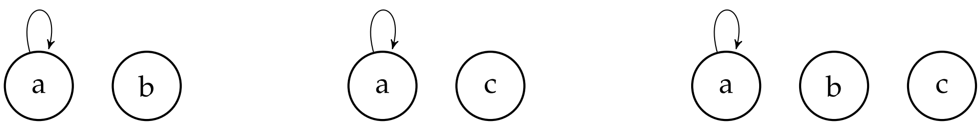


Figure 23: Argument graphs W (left), Y (middle), and $W + Y$ (right).

Set(W)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
∅	✓	✓	×	×	×	×	×
{a}	×	×	×	×	×	×	×
{b}	✓	✓	✓	✓	✓	×	✓

Table 25: Semantics of argument graph W.

Set(Y)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
∅	✓	✓	×	×	×	×	×
{a}	×	×	×	×	×	×	×
{c}	✓	✓	✓	✓	✓	×	✓

Table 26: Semantics of argument graph Y.

[4]This result in itself is contradictory, as the Distance-Based Measure cannot exceed 1.

Set($W+Y$)	Conflict-free	Admissible	Complete	Preferred	Grounded	Stable	Naive
∅	✓	✓	×	×	×	×	×
{a}	×	×	×	×	×	×	×
{b}	✓	✓	×	×	×	×	×
{c}	✓	✓	×	×	×	×	×
{b, c}	✓	✓	✓	✓	✓	×	✓

Table 27: Semantics of argument graph $W+Y$.

- **(I_p)** Consider argument graph A in Figure 1, B in Figure 2, and the merged graph $A+B$, equivalent to graph D in Figure 2 as a counterexample. Their semantics are available in Table 10, 1, and 16, respectively. $I_p(A) = 2+0-1 = 1$, $I_p(B) = 1+0-1 = 0$, and $I_p(A+B) = 1+0-1 = 0$. It is apparent that $I_p(A+B) = 0 \ngeq 1 = 1+0 = I_p(A) + I_p(B)$.

- **(I_{con})** Analyze the following counterexample with argument graph C' in Figure 3 and L' in Figure 7. $I_{con}(C') = 1$ and $I_{con}(L') = 2$. For the joint graph $C' + L'$ it holds that $I_{con}(C' + L') = 2$ s.t. $I_{con}(C' + L') = 2 \ngeq 3 = 1 + 2 = I_{con}(C') + I_{con}(L')$.

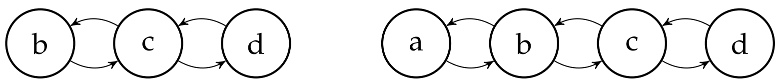


Figure 24: Argument graphs Z (left) and $A+Z$ (right).

- **(I_{ind})** Observe the following counterexample, with two graphs A in Figure 1 and Z in Figure 24. The joint graph $A+Z$ is visible in Figure 24. $I_{ind}(A) = 2$ and $I_{ind}(Z) = 3$. However, $I_{ind}(A+Z) = 4$. Clearly, $I_{ind}(A+Z) = 4 \ngeq 5 = 2+3 = I_{ind}(A) + I_{ind}(Z)$.

- **(I_{db})** Take into account the following counterexample with graph A in Figure 1, B in Figure 2 and their joint graph $A+B$, which is equivalent to graph D in Figure 2. In Example 3.28 it was computed that $I_{db}(A) = 0.75$ and in Example 3.26 it was calculated that $I_{db}(D) = \frac{5}{9} \approx 0.56$. We calculate $I_{db}(B)$: $GD(B) = 1+(3\times 3) = 10$. As $n = 2$, $\max = 2^2 \times (2+1) = 12$ and $\min = 2^2 = 4$.

$$I_{db}(B) = \frac{\max - GD(B)}{\max - \min} = \frac{12-10}{12-4} = \frac{1}{4} = 0.25$$

$$I_{db}(A+B) \approx 0.56 \ngeq 1 = 0.75 + 0.25 = I_{db}(A) + I_{db}(B).$$

□

4.2.5 Cyclicaltiy Postulate Propositions and Proofs

Proposition 4.13 *Reinforcement* is satisfied by I_{wco} but not by I_{pr}, I_{ngr}, I_{nst}, I_{dr}, I_{win}, I_{wout}, I_{cc}, and I_{wcc}.

Proof.

- **(I_{wco})** Consider two graphs G and G' which satisfy all the criteria of the Reinforcement principle as defined in Section 4.1.6. Recall that I_{wco} is defined as the following where $|X|$ is the number of nodes in a component X in the graph G.

$$I_{wco}(G) = \sum_{X \in \text{Components}(G)} (|X| - 1)^2$$

First, consider graph G in which the nodes are paired as (a_i, b_i) where $i \in \{0, ..., n-1\}$ and $n \geq 3$ according to Reinforcement. This means that only two types of nodes exist in G: pairs of nodes and individual disconnected nodes. The contribution from each paired component in the graph is $(|X| - 1)^2 = (2-1)^2 = 1$. For the disconnected nodes the WeightedComponentCount is always zero, because $(|X| - 1)^2 = (1-1)^2 = 0$. The total value of $I_{wco}(G)$ is expressed as the summation of 1's proportional to the number of paired components in G. In terms of n, the WeightedComponentCount in G is:

$$I_{wco}(G) = \sum_{i=0}^{n-1} 1 = n.$$

Now observe graph G' with the attack pattern $(a_i, a_i + 1)$ where $i \in \{0, ..., n-1\}$, as determined by the Reinforcement postulate. With the attack relation, it is ensured, that the graph contains one large component with $n+1$ nodes that are connected and form one component, while the rest of nodes are disconnected nodes. As mentioned above, the disconnected nodes do not contribute because they amount to 0. The contribution from the larger component in G', with $n+1$ nodes, is $(n+1-1)^2 = n^2$. As a result the WeightedComponentCount for G' is:

$$I_{wco}(G') = n^2.$$

Since $I_{\text{wco}}(G) = n < n^2 = I_{\text{wco}}(G')$ always holds, Reinforcement is fulfilled.

- **(I_{pr})** If for two graphs G and G' the conditions of Reinforcement are satisfied, G and G' must be acyclic. Recall that I_{pr} counts the number of preferred extensions and subtracts one. Since the graphs G and G' are acyclic and every acyclic graph has exactly one preferred extension [17] [15], it follows that for any graph that satisfies the criteria of Reinforcement, $I_{\text{pr}}(G) = I_{\text{pr}}(G') = 1 - 1 = 0$ s.t. $I_{\text{ngr}}(G) = 0 \nless 0 = I_{\text{ngr}}(G')$.

For the following proofs consider graphs T in Figure 12 and T' in Figure 13 as counterexamples which fulfill the requirements of Reinforcement.

- **(I_{ngr})** Since Disjoint Additivity holds for I_{ngr}, the graphs T and T' can be divided into their disjoint components. First observe graph T which consists of three pairs of nodes isomorphic to graph B in Figure 2 and two disconnected nodes isomorphic to graph M in Figure 8. We have already calculated $I_{\text{ngr}} = 0$ for graphs B and M in Examples 3.5 and 3.4, respectively, s.t.

$$\begin{aligned} I_{\text{ngr}}(T) = & \sum_{i\in\{0,1,2\}} I_{\text{ngr}}((\{a_i, b_i\}, \{(a_i, b_i)\})) + I_{\text{ngr}}((\{a_3\}, \emptyset)) + I_{\text{ngr}}((\{b_3\}, \emptyset)) \\ = & (3 \times 0) + 0 + 0 = 0. \end{aligned}$$

  Graph T' also consists of disconnected nodes and a larger component consisting of four nodes,

$$((\{a_0, a_1, a_2, a_3\}, \{(a_0, a_1), (a_1, a_2), (a_2, a_3)\})),$$

  which we label T'_4 for readability. Given, that for the disconnected nodes $I_{\text{ngr}} = 0$, we calculate $I_{\text{ngr}}(T'_4)$, based on the semantics in Table 28, followed by $I_{\text{ngr}}(T')$.

$$\begin{aligned} I_{\text{ngr}}(T'_4) &= |\text{Nodes}(T'_4) \setminus (\text{Extension}_g(T'_4) \cup \text{Attacked}(T'_4))| \\ &= |\{a_0, a_1, a_2, a_3\} \setminus (\{a_0, a_2\} \cup \{a_1, a_3\})| \\ &= |\{a_0, a_1, a_2, a_3\} \setminus \{a_0, a_1, a_2, a_3\}| = |\emptyset| = 0 \end{aligned}$$

$$\begin{aligned} I_{\text{ngr}}(T') &= \sum_{i\in\{0,1,2,3\}} I_{\text{ngr}}((\{b_i\}, \emptyset)) + I_{\text{ngr}}(T'_4) \\ &= 0 + (4 \times 0) = 0 \end{aligned}$$

  In total, $I_{\text{ngr}}(T) = 0 \not< 0 = I_{\text{ngr}}(T')$.

Set(T'_4)	**Conflict-free**	**Admissible**	**Complete**	**Preferred**	**Grounded**	**Stable**	**Naive**
$\emptyset$	✓	✓	×	×	×	×	×
$\{a_0\}$	✓	✓	×	×	×	×	×
$\{a_1\}$	✓	×	×	×	×	×	×
$\{a_2\}$	✓	×	×	×	×	×	×
$\{a_3\}$	✓	×	×	×	×	×	×
$\{a_0, a_2\}$	✓	✓	✓	✓	✓	✓	✓
$\{a_0, a_3\}$	✓	×	×	×	×	×	✓
$\{a_1, a_3\}$	✓	×	×	×	×	×	✓

Table 28: Semantics of graph T'_4.

- **(I_{nst})** As Disjoint Additivity holds for I_{nst}, T and T' are divided into their components to facilitate calculations. Graph T consists of three pairs of nodes, isomorphic to B for which we have calculated $I_{\text{nst}} = 0$ in Example 3.8. T further consists of two disconnected nodes, where under the proof for I_{nst} in Proposition 4.6, we have computed that $I_{\text{nst}} = 0$ for a disconnected node. We calculate $I_{\text{nst}}(T)$.

$$\begin{aligned} I_{\text{nst}}(T) &= \sum_{i \in \{0,1,2\}} I_{\text{nst}}((\{a_i, b_i\}, \{(a_i, b_i)\})) + I_{\text{nst}}((\{a_3\}, \emptyset)) + I_{\text{nst}}((\{b_3\}, \emptyset)) \\ &= (3 \times 0) + 0 + 0 = 0 \end{aligned}$$

Graph T' also consists of disconnected nodes and a larger component consisting of four nodes,

$$((\{a_0, a_1, a_2, a_3\}, \{(a_0, a_1), (a_1, a_2), (a_2, a_3)\})),$$

which we have labeled T'_4 as above. Given, that for the disconnected nodes $I_{\text{nst}} = 0$, we calculate $I_{\text{nst}}(T'_4)$, followed by $I_{\text{nst}}(T')$.

$$N = \emptyset : \text{Extension}_s(\text{Induced}(T'_4, \emptyset)) = \text{Extension}_s(\{a_0, a_1, a_2, a_3\}) = \{a_0, a_2\}$$

Since $N = \emptyset$ already results in a stable extension $\{a_0, a_2\}$, $I_{\text{nst}}(T'_4) = 0$.

$$\begin{aligned} I_{\text{nst}}(T') &= \sum_{i \in \{0,1,2,3\}} I_{\text{nst}}((\{b_i\}, \emptyset)) + I_{\text{nst}}(T'_4) \\ &= (4 \times 0) + 0 = 0 \end{aligned}$$

For both graphs T and T' no arguments have to be removed to obtain a stable extension. Thus, $I_{\text{nst}}(T) = 0 \not< 0 = I_{\text{nst}}(T')$.

- **(I_{dr})** Both T and T' contain attacks s.t. $I_{\text{dr}}(T) = 1 \not< 1 = I_{\text{dr}}(T')$.

- **(I_{win})** $I_{\text{win}}(T) = 3 \times 1 = 3$. $I_{\text{win}}(T') = 3 \times 1 = 3$. Consequently, $I_{\text{win}}(T) = 3 \not< 3 = I_{\text{win}}(T')$.

- **(I_{wout})** $I_{\text{wout}}(T) = 3 \times 1 = 3$ and $I_{\text{wout}}(T') = 3 \times 1 = 3$. It is visible that $I_{\text{wout}}(T) = 3 \not< 3 = I_{\text{wout}}(T')$.

- **(I_{cc} and I_{wcc})** Both graphs T and T' are acyclic as required by the reinforcement principle. Consequently, $I_{\text{cc}}(T) = I_{\text{cc}}(T') = 0$. Then $I_{\text{cc}}(T) = 0 \not< 0 = I_{\text{cc}}(T')$. Similarly, $I_{\text{wcc}}(T) = I_{\text{wcc}}(T') = 0$. Thus $I_{\text{wcc}}(T) = 0 \not< 0 = I_{\text{wcc}}(T')$. □

Proposition 4.14 I_{ngr}, I_{cc}, and I_{wcc} satisfy *Cycle Precedence* while the measures I_{pr}, I_{nst}, I_{dr}, I_{in}, I_{win}, I_{wout}, and I_{wco} do not.

Proof.

- **(I_{ngr})** Consider an acyclic graph $G = (\mathcal{A}, \mathcal{R})$. Since G is acyclic, there are no cycles of attacks, meaning that every argument is either in the grounded extension, $\text{Extension}_g(\text{G})$, or attacked by some argument in the grounded extension. Suppose that there exists an argument $a \in \mathcal{A} \setminus \text{Extension}_g(\text{G})$ that is not attacked by any member of $\text{Extension}_g(\text{G})$. This leads to the following cases:
  - If a is unattacked, it must belong to $\text{Extension}_g(\text{G})$, contradicting $a \notin \text{Extension}_g(\text{G})$.
  - If a is attacked, the attacker b must either be
    - in $\text{Extension}_g(\text{G})$, contradicting our statement, or
    - outside $\text{Extension}_g(\text{G})$, which also is contradicting because argument b would be defended by $\text{Extension}_g(\text{G})$, indicating that a is defended by $\text{Extension}_g(\text{G})$.

  As both cases lead to contradictions, no such a exists. Consequently, in an acyclic graph, the number of arguments not in $\text{Extension}_g(\text{G})$ and not attacked by $\text{Extension}_g(\text{G})$ is always 0. Now consider an elementary cycle G'. Given that G' is an elementary cycle there are no unattacked arguments. This in turn means that the empty set always satisfies the conditions for being a complete extension. Since the grounded extension is the minimal (w.r.t. $\subseteq$) complete extension, it follows automatically that $\emptyset$ is the unique grounded extension in every elementary cycle. Given that $\text{Extension}_g(G') = \emptyset$, we compute $I_{\text{ngr}}(G')$:

  $$\begin{aligned} I_{\text{ngr}}(G') &= |\text{Nodes}(G') \setminus (\text{Extension}_g(G') \cup \text{Attacked}(G'))| \\ &= \left|\{\text{Nodes}(G')\} \setminus (\emptyset \cup \emptyset)\right| \\ &= \left|\{\text{Nodes}(G')\} \setminus \emptyset\right| = |\text{Nodes}(G')|. \end{aligned}$$

  It follows that $I_{\text{ngr}}(G) = |\text{Nodes}(G)| \geq 1$. It holds that $I_{\text{ngr}}(G) = 0 < 1 \leq I_{\text{ngr}}(G')$ for an acyclic graph G and an elementary cycle G'.

- **(I_{cc})** By definition, it holds that for an acyclic graph G, $I_{\text{cc}}(G) = 0$ and for a graph G' consisting of an elementary cycle $I_{\text{cc}}(G') = 1$. So, $I_{\text{cc}}(G) = 0 < 1 = I_{\text{cc}}(G')$ applies.

- **(I_{wcc})** For an acyclic graph G, $I_{\text{wcc}}(G) = 0$. For a graph G' consisting of an elementary cycle with n nodes where $n \in \mathbb{Z}$, $n \geq 1$, $I_{\text{wcc}}(G') = \frac{1}{n}$. It follows that $I_{\text{wcc}}(G) = 0 < \frac{1}{n} = I_{\text{wcc}}(G')$.

- **(I_{pr})** Consider acyclic graph B in Figure 2 and its semantics in Table 1 and an elementary cycle N in Figure 8 and its semantics in Table 7 as a counterexample. $I_{\text{pr}}(N) = 1 - 1 = 0 \not< 0 = 1 - 1 = I_{\text{pr}}(B)$.

- **(I_{nst})** Consider the acyclic graph B in Figure 2 and the elementary cycle H in Figure 3 as a counterexample. As calculated in Examples 3.8 and 3.9, $I_{\text{nst}}(B) = 0$ and $I_{\text{nst}}(H) = 1$, respectively, s.t. $I_{\text{nst}}(B) = 0 \not< 1 = I_{\text{nst}}(H)$.

- **(I_{dr})** Consider the acyclic graph B in Figure 2 and elementary cycle H in Figure 3. $I_{\text{dr}}(J) = 1 \not< 1 = I_{\text{dr}}(H)$.

- **(I_{win}, I_{wout}, and I_{wco})** Observe these counterexamples with the acyclic graph J in Figure 4 and elementary cycle A in Figure 1. $I_{\text{win}}(J) = 1 + 1 = 2 \not< 2 = 1 + 1 = I_{\text{win}}(A)$, $I_{\text{wout}}(J) = 1 + 1 = 2 \not< 2 = 1 + 1 = I_{\text{wout}}(A)$, and $I_{\text{wco}}(J) = (3-1)^2 = 4 \not< 1 = (2-1)^2 = I_{\text{wco}}(A)$. □

Proposition 4.15 *Size Sensitivity* is fulfilled by I_{wcc} but not by I_{pr}, I_{ngr}, I_{nst}, I_{dr}, I_{win}, I_{wout}, I_{cc}, and I_{wco}.

Proof.

- **(I_{wcc})** Consider two elementary cycles $G = (\mathcal{A}, \mathcal{R})$ and $G' = (\mathcal{A}', \mathcal{R}')$ where $|\mathcal{A}'| < |\mathcal{A}|$. $I_{\text{wcc}}(G) = \sum_{C \in \text{Cycles}(G)} \frac{1}{|C|}$ where $|C| \geq 1$ is the number of nodes involved in the cycle C. Thus, $I_{\text{wcc}}(G) = \frac{1}{|\mathcal{A}|}$ and $I_{\text{wcc}}(G') = \frac{1}{|\mathcal{A}'|}$. Since $|\mathcal{A}'| < |\mathcal{A}|$, it follows that $I_{\text{wcc}}(G) = \frac{1}{|\mathcal{A}|} < \frac{1}{|\mathcal{A}'|} = I_{\text{wcc}}(G')$.

- **(I_{pr})** Analyze graph N in Figure 8 and graph H in Figure 3, and their semantics in Table 7 and 12, respectively. Both graphs are elementary cycles s.t. $|\text{Nodes}(N)| = 1 < 3 = |\text{Nodes}(H)|$. However, $I_{\text{pr}}(N) = 1 - 1 = 0 \not> 0 = 1 - 1 = I_{\text{pr}}(H)$.

Unless mentioned otherwise, for the following proofs observe graph A in Figure 1 and graph H in Figure 3. Both graphs are elementary cycles and $|\text{Nodes}(A)| = 2 < 3 = |\text{Nodes}(H)|$. The semantics of H are in Table 12.

- **(I_{ngr})** As computed in Example 3.6 $I_{\text{ngr}}(A) = 2$. We calculate $I_{\text{ngr}}(H)$:

$$\begin{aligned} I_{\text{ngr}}(H) &= |\text{Nodes}(H) \setminus (\text{Extension}_g(H) \cup \text{Attacked}(H))| \\ &= |\{e, f, g\} \setminus (\emptyset \cup \emptyset)| \\ &= |\{e, f, g\} \setminus \emptyset| = |\{e, f, g\}| = 3. \end{aligned}$$

  As visible, $I_{\text{ngr}}(A) = 2 \not> 3 = I_{\text{ngr}}(H)$.

- **(I_{nst})** In Example 3.9 we have shown that $I_{\text{nst}}(H) = 1$. For graph N in Figure 8, whose semantics are in Table 7, we compute that $I_{\text{nst}}(N) = 1$.

$$N = \emptyset : \text{Extension}_s(\text{Induced}(N, \emptyset)) = \text{Extension}_s(\{a\}) = \emptyset$$
$$N = \{a\} : \text{Extension}_s(\text{Induced}(N, \{a\})) = \text{Extension}_s(\emptyset) = \emptyset$$

It holds that $I_{\text{nst}}(N) = 1 \not> 1 = I_{\text{nst}}(H)$.

- **(I_{dr})** $I_{\text{dr}}(A) = 1 \not> 1 = I_{\text{dr}}(H)$.
- **(I_{win})** $I_{\text{win}}(A) = 2 \times 1 = 2 \not> 3 = 3 \times 1 = I_{\text{win}}(H)$.
- **(I_{wout})** $I_{\text{wout}}(A) = 2 \times 1 = 2 \not> 3 = 3 \times 1 = I_{\text{wout}}(H)$.
- **(I_{cc})** $I_{\text{cc}}(A) = 1 \not> 1 = I_{\text{cc}}(H)$.
- **(I_{wco})** $I_{\text{wco}}(A) = (2-1)^2 = 1 \not> 4 = (3-1)^2 = I_{\text{wco}}(H)$. □

5 Conclusion

The objective of this Bachelor thesis was to give an overview of the inconsistency measures in argument graphs in abstract argumentation proposed in literature [24] [1] until now, while offering a deeper and unexplored perspective of their characteristics through descriptive evaluation and the utilization of rationality postulates. In regards to descriptive evaluation, we provided detailed explanations for 14 inconsistency measures, and considered their strengths and weaknesses. As for the evaluation based on rationality postulates, we examined 18 rationality postulates [24] [1] [39] [40], six of which, namely Penalty, Free-Node Dilution, p-Exchange, g-Exchange, s-Exchange, and $naive$-Exchange, we formulated in the context of abstract argumentation, inspired by [35], [28] as well as, [9] and [30], respectively. Finally, for inconsistency measures, whose compliance with the rationality postulates had not been previously established, we presented proofs to demonstrate the satisfaction or violation with these postulates. We summarized our findings in Table 15 and discussed them, including those that have been showed before in [24], [1], [29], and [38].

In the following sections we briefly summarize our results and point to directions towards possible future work.

5.1 Summary of Results

Results of the descriptive evaluation show that while in general, the extension-based measures provide a structured, semantics-driven evaluation of inconsistency, the strengths and weaknesses of these measures especially vary depending on the chosen semantics, as different semantics capture distinct aspects of argumentation.

Likewise, the strengths and weaknesses of the graph structure measures depend on which structural aspects they focus on. Measures that emphasize cycles such as CycleCount or WeightedCycleCount, for instance, only focus on circular inconsistencies to different extents and may overlook other structural issues. Despite the different focuses of all the measures, the Distance-based Measure stands out as an especially fine-granular measure. This is because it considers direct and indirect attacks, incorporates every node and arc into the measure including disconnected nodes, is normalized ranging from 0 to 1, where 0 and 1 are exclusivity reserved for a consistent and a maximally inconsistent graph, respectively, and reserves a certain range of values for cyclic and acyclic graphs.

Concerning the fulfillment of principles, our findings are summarized in Table 15 and discussed in detail below it, in Section 4.2. Overall, it is apparent that the compliance or noncompliance for each inconsistency measure varies. However, all inconsistency measures adhere to Consistency, Isomorphic Invariance and Free-Node Dilution. It is also visible that the Distance-Based Measure is the only measure that satisfies Reinforcement, Cycle Precedence and Size Sensitivity, simultaneously as well as Contradiction. Conversely, Freeness is satisfied by all measures except for the Distance-Based Measure. For the principles related to strong equivalence, each extension-based measure focusing on semantics σ, where σ is preferred, grounded, stable, or naive, satisfies its respective σ-Exchange. Nevertheless, other measures also satisfy certain σ-Exchange postulates, including the two graph structure-based measures, WeightedComponentCount and In-degree Measure, which satisfy σ-Exchange under grounded and naive semantics, respectively.

The findings in this thesis also entail that the satisfaction of many rationality postulates is not a sufficient criterion for evaluating an inconsistency measure. To illustrate why a comparison based solely on the number of satisfied postulates is not a valid assessment, consider the Drastic and Distance-based Measure, which each satisfy 11 and 10, respectively, out of 18 principles. However, Drastic is a simple binary measure, while the Distance-based measure is a very fine-granular measure, as described above. This implies that Drastic's compliance with most postulates is based on the coarseness of the measure rather than its features. This example also highlights that considering only adherence to rationality postulates, without assessing the measure's strengths and weaknesses, omits critical insights. Additionally, a quantitative assessment of satisfied postulates would disregard the different natures of the principles. It would be misleading to consider the satisfaction of postulates like Consistency and Normalization equivalent to the fulfillment of principles that focus on strong equivalence.

In Summary, we conclude that the strengths and weaknesses and the compliance or noncompliance of the inconsistency measures with the rationality postulates tend to vary, as each measure captures different aspects of inconsistency in argument graphs to various extents. We emphasize that this diverse nature does not mean that a single measure is more superior than another; rather, their usefulness depends on the specific context and the properties being analyzed. This highlights the importance of considering multiple measures to gain a comprehensive understanding of inconsistency within argument graphs. While we reformulated six postulates for inconsistency measurements in abstract argumentation graphs, these observations highlight the need for further exploration of rationality postulates in inconsistency measurement.

5.2 Future Work

So far, only limited work exists on inconsistency measures in abstract argument graphs, namely [24] and [1]. As such, many areas of investigation are still open, including the development of further inconsistency measures in abstract argument graphs. To gain more perspectives of the characteristics of inconsistency measures, future work also may consist of the consideration and proposition of more rationality postulates for abstract argumentation graphs to assess the inconsistency measures in this thesis even more comprehensively.

However, as more measures and postulates are proposed, Thimm [39] warns that more different measures yield different assessments of how inconsistent an argument graph is, depending on which rationality postulates they satisfy or violate, creating ambiguity in what it means for an argument graph to be more or less inconsistent. One area of extension therefore could be the identification of a minimal and natural set of postulates that uniquely define a good inconsistency measure [39]. In the context of this thesis, one starting approach would be taking into account the mutual exclusivity of the rationality postulates. The identification of which postulates cannot hold together would allow for the definition of classes of inconsistency measures, leading to a minimal foundational set of postulates to make inconsistency measures more rigorous.

Moreover, future work may incorporate an analysis of the computational complexity of the inconsistency measures listed in this thesis. Prior studies such as [43], [39], [42], and [44], have explored the computational complexity of other inconsistency measures in different contexts, providing important information about the practicality and extent of applicability of the measure by revealing whether the computation of inconsistency measures can be performed efficiently [31].

Another direction of future work may involve capturing the expressivity of the inconsistency measures in this thesis, as in [37] and [39]. Expressivity quantifies to which extent a given inconsistency measure is capable to distinguish between different inconsistent graphs, complementing rationality postulates that tend to focus on single aspects [37].

References

[1] L. Amgoud and J. Ben-Naim. Measuring disagreement in argumentation graphs. In S. Moral, O. Pivert, D. Sánchez, and N. Marín, editors, *Scalable Uncertainty Management*, pages 1–14, Cham, October 2017. Springer International Publishing.

[2] L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3):413–436, November 2009.

[3] P. Baroni and M. Giacomin. Solving semantic problems with odd-length cycles in argumentation. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Lecture Notes in Computer Science*, 2711:440–451, April 2004.

[4] R. Baumann and G. Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In P. Baroni, F. Cerutti, M. Giacomin, and G. R. Simari, editors, *Proceedings of the 3rd Conference on Computational Models of Argument (COMMA 2010)*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 75–86. IOS Press, January 2010.

[5] R. Baumann, G. Brewka, and M. Ulbricht. Revisiting the foundations of abstract argumentation – semantics based on weak admissibility and weak defense. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:2742–2749, April 2020.

[6] R. Baumann, W. Dvořák, T. Linsbichler, and S. Woltran. A general notion of equivalence for abstract argumentation. *Artificial Intelligence*, 275:1–53, June 2019.

[7] T. Bench-Capon. Dilemmas and paradoxes: cycles in argumentation frameworks. *Journal of Logic and Computation*, 26:1055–1064, January 2014.

[8] T. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. volume 171, pages 619–641, January 2007. Argumentation in Artificial Intelligence.

[9] P. Besnard. Revisiting postulates for inconsistency measures. In E. Fermé and J. Leite, editors, *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14)*, pages 383–396. Springer International Publishing, Cham, September 2014.

[10] P. Besnard, A. Garcia, A. Hunter, S. Modgil, H. Prakken, G. Simari, and F. Toni. Introduction to structured argumentation. *Argument & Computation*, 5:1–4, February 2014.

[11] P. Besnard and J. Grant. Relative inconsistency measures. *Artificial Intelligence*, 280:103231, March 2020.

[12] V. Beuselinck, J. Delobelle, and S. Vesic. A principle-based account of self-attacking arguments in gradual semantics. *Journal of Logic and Computation*, 33(2):230–256, February 2023.

[13] G. Bodanza and F. Tohmé. Two approaches to the problems of self-attacking arguments and general odd-length cycles of attack. *Journal of Applied Logic*, 7:403–420, December 2009.

[14] A. Bondarenko, P. Dung, R. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1):63–101, April 1997.

[15] W. Chen. Collective argumentation: The case of aggregating support-relations of bipolar argumentation frameworks. *Electronic Proceedings in Theoretical Computer Science*, 335:87–102, June 2021.

[16] R. Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer Berlin, Heidelberg, 6 edition, January 2025.

[17] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, September 1995.

[18] B. Fazzinga and R. Mellema. Argumentation in ai. In M. Chetouani, V. Dignum, P. Lukowicz, and C. Sierra, editors, *Human-Centered Artificial Intelligence: Advanced Lectures*, pages 347–362. Springer International Publishing, Cham, April 2023.

[19] S. A. Gaggl and S. Woltran. Strong equivalence for argumentation semantics based on conflict-free sets. In W. Liu, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 38–49. Springer Berlin Heidelberg, Berlin, Heidelberg, June 2011.

[20] J. Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19:435–444, July 1978.

[21] D. Grinberg. *An introduction to graph theory*. November 2024.

[22] E. Hadoux, A. Hunter, and S. Polberg. Strategic argumentation dialogues for persuasion: Framework and experiments based on modelling the beliefs and concerns of the persuadee. *Argument & Computation*, 14(2):109–161, 2023.

[23] W. Hochstättler, D. Andres, R. Nickel, and M. Schulte. *Algorithmische Mathematik*. FernUniversität in Hagen, Fakultät für Mathematik und Informatik, Hagen, Germany, 2020.

[24] A. Hunter. Measuring inconsistency in argument graphs. August 2017.

[25] A. Hunter and S. Konieczny. Shapley inconsistency values. *Proceedings, 10th International Conference on Principles of Knowledge Representation and Reasoning, KR 2006*, pages 249–259, January 2006.

[26] L. Johnston and M. Wootters. Lecture 10: Connected components of undirected and directed graphs. Stanford University CS161 Lecture Notes, October 2017.

[27] D. Martinez, A. Garcia, and G. Simari. An abstract argumentation framework with varied-strength attacks. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 135–144. AAAI Press, January 2008.

[28] K. Mu, W. Liu, Z. Jin, and D. Bell. A syntax-based approach to measuring the degree of inconsistency for belief bases. *International Journal of Approximate Reasoning*, 52(7):894–905, October 2011.

[29] I. Müßig. Inconsistency measures in abstract argumentation in argument graphs. Technical report, FernUniversität in Hagen, Hagen, Germany, August 2024. Seminar paper presented at the Bachelorseminar Künstliche Intelligenz.

[30] E. Oikarinen and S. Woltran. Characterizing strong equivalence for argumentation frameworks. *Artificial Intelligence*, 175(14):1985–2009, 2011.

[31] C. H. Papadimitriou. *Computational complexity*, page 260–265. John Wiley and Sons Ltd., 2003.

[32] T. Rienstra, C. Sakama, L. van der Torre, and B. Liao. A principle-based robustness analysis of admissibility-based argumentation semantics. *Argument & Computation*, 11:305–339, November 2020.

[33] A. Sadiq, H. Abdulah, and A. Kareem. Argumentation frameworks – a brief review. *International Journal of Online and Biomedical Engineering (iJOE)*, 18:55–70, February 2022.

[34] K. Takahashi. Odd or even: Handling n-lemmas in a dynamic argumentation framework. In *SAFA'22: Fourth International Workshop on Systems and Algorithms for Formal Argumentation 2022*, volume 3236, Cardiff, Wales, United Kingdom, September 2022. CEUR Workshop Proceedings.

[35] M. Thimm. Measuring inconsistency in probabilistic knowledge bases. In J. Bilmes and A. Ng, editors, *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI'09)*, pages 530–537. AUAI Press, June 2009.

[36] M. Thimm. Inconsistency measures for probabilistic logics. *Artificial Intelligence*, 197:1–24, April 2013.

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

[38] M. Thimm. On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. *KI - Künstliche Intelligenz*, 31(1):31–39, March 2017.

[39] M. Thimm. On the evaluation of inconsistency measures. In J. Grant and M. V. Martinez, editors, *Measuring Inconsistency in Information*, volume 73 of *Studies in Logic*. College Publications, February 2018.

[40] M. Thimm. Inconsistency measurement. In *Scalable Uncertainty Management: 13th International Conference, SUM 2019, Compiègne, France*, pages 1–4, December 2019.

[41] M. Thimm, F. Cerutti, and T. Rienstra. Probabilistic graded semantics. In S. Modgil, K. Budzynska, and J. Lawrence, editors, *Computational Models of Argument - Proceedings of COMMA 2018*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 369–380, Warsaw, Poland, September 2018. IOS Press.

[42] M. Thimm and J. P. Wallner. Some complexity results on inconsistency measurement. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, Cape Town, South Africa, April 2016.

[43] M. Thimm and J. P. Wallner. On the complexity of inconsistency measurement. *Artificial Intelligence*, 275:411–456, October 2019.

[44] M. Ulbricht, M. Thimm, and G. Brewka. Measuring strong inconsistency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018.

[45] M. Ulbricht and J. Wallner. Strong explanations in abstract argumentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:6496–6504, May 2021.

[46] F. H. van Eemeren, R. Grootendorst, A. F. Snoeck Henkemans, J. A. Blair, R. H. Johnson, E. C. W. Krabbe, C. Plantin, D. N. Walton, and C. A. Willard. *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments*. Lawrence Erlbaum Associates, Mahwah, New Jersey, United States of America, 1996.

[47] D. Walton. *Methods of Argumentation*. Cambridge University Press, United States of America, January 2010.