

Ranking Arguments by Minimal Attack Deletion for Grounded Semantics

Master's Thesis

in partial fulfilment of the requirements for
the degree of *Master of Science (M.Sc.)*
in Informatik

submitted by
Jakob Haastert

First examiner: Prof. Dr. Matthias Thimm
Artificial Intelligence Group

Advisor: Dr. Kenneth Skiba
Artificial Intelligence Group

Statement

I declare that I have written the master's thesis independently and without unauthorized use of third parties. I have only used the indicated resources and I have clearly marked the passages taken verbatim or in the sense of these resources as such. The assurance of independent work also applies to any drawings, sketches or graphical representations. The work has not previously been submitted in the same or similar form to the same or another examination authority and has not been published. By submitting the electronic version of the final version of the master's thesis, I acknowledge that it will be checked by a plagiarism detection service to check for plagiarism and that it will be stored exclusively for examination purposes.

I explicitly agree to have this thesis published on the webpage of the artificial intelligence group and endorse its public availability.

Software created for this work has been made available as open source; a corresponding link to the sources is included in this work. The same applies to any research data.

Koblenz, 28.06.2026

(Place, Date)


(Signature)

Zusammenfassung

Rangbasierte Semantiken bewerten die Stärke von Argumenten in der abstrakten Argumentation. Diese Arbeit führt eine Semantik ein, die Argumente einordnet, indem sie die minimale Anzahl von Angriffs-Löschungen berechnet, die erforderlich ist, um ihre Aufnahme in die Grounded Extension zu erzwingen. Wir evaluieren diesen Ansatz anhand etablierter Prinzipien, um seine Stärken und Grenzen zu identifizieren. Um eine mangelnde Differenzierung zwischen akzeptierten Argumenten zu beheben, schlagen wir eine Modifikation vor, die die Granularität erhöht. Abschließend implementieren wir beide Semantiken mittels Answer Set Programming und bewerten ihre praktische Nutzbarkeit.

Abstract

Ranking-based semantics evaluate the strength of arguments in abstract argumentation. This thesis introduces a semantics that ranks arguments by calculating the minimum number of attack deletions required to enforce their inclusion in the grounded extension. We evaluate this approach against established principles to identify its strengths and limitations. To resolve a lack of differentiation among accepted arguments, we propose a modification that increases granularity. Finally, we implement both semantics using Answer Set Programming and assess their computational feasibility.

Contents

1	Introduction	1
2	Background	3
2.1	Abstract Argumentation	3
2.2	Extension-Based Semantics	4
2.3	Ranking-Based Semantics	5
2.4	Implementation Preliminaries	6
3	Attack Deletion Ranking-Based Semantics	9
3.1	Attack Deletion Scoring Function	9
3.2	Attack Deletion Ranking-Based Semantics	9
4	Principle Analysis	11
4.1	Abstraction	11
4.2	Independence	12
4.3	Total	13
4.4	Non-Attacked Equivalence	13
4.5	Void Precedence	14
4.6	Cardinality Precedence	15
4.7	Quality Precedence	15
4.8	Self-Contradiction	16
4.9	Counter-Transitivity	17
4.10	σ -Compatibility	19
4.11	Defense Precedence	21
4.12	Distributed-Defense Precedence	22
4.13	Change-Related Principles	23
4.14	Comparison Between Ranking-Based Semantics	26
4.15	Interpretation and Discussion of Results	28
5	Extended Attack Deletion Ranking-Based Semantics	29
5.1	Principle Analysis	33
5.1.1	Void Precedence	33
5.1.2	Abstraction	34
5.1.3	Independence	35
5.1.4	Total	36
5.1.5	Non-Attacked Equivalence	36
5.1.6	Counter-Transitivity	36
5.1.7	Cardinality Precedence	37
5.1.8	<i>grounded</i> -Compatibility	37
5.1.9	Addition of Attack Branch	38
5.1.10	Attack vs Full Defense	38
5.2	Improvement over AttDel	38

6	Implementation	39
6.1	Primitive Solution	39
6.2	ASP Solution	40
6.2.1	ASP Implementation using the Grounded Extension	40
6.2.2	ASP Implementation using Strongly Admissible Sets	42
6.2.3	Search Space Optimization	44
6.2.4	Implementing ModAttDel	45
6.2.5	Performance	45
7	Conclusion	48
7.1	Limitations	50
7.2	Future Work	51

1 Introduction

Abstract argumentation offers a powerful method of knowledge representation and reasoning in Artificial Intelligence (AI). Its simple structure of arguments and attacks can represent complex argumentation scenarios in a simple directed graph [9]. Various semantics have been developed to determine whether sets of arguments are acceptable or not [9]. These traditional semantics, such as the grounded, preferred, complete, and stable extensions, provide a binary qualitative assessment on the acceptability of arguments [9]. This, however, can reach its limits in real-world scenarios, where there are large amounts of arguments to be evaluated and relatively few extensions to classify them by. Ranking-based semantics allow for a more customized and finer evaluation of arguments [1]. Depending on the use case, different semantics can be used to rank arguments based on various criteria [1] which are established by evaluating different principles from the literature [6]. The grounded extension is highly intuitive because it identifies the arguments present in every complete extension of a framework [9], implying a high level of acceptability. However, its binary classification lacks nuance. It treats an almost acceptable argument exactly the same as a completely defeated one. To solve this, we introduce an approach that preserves the logical foundation of the grounded extension while adding the granularity of a ranking system. By measuring the effort required, specifically the number of attack deletions to introduce a rejected argument into the grounded extension, we can create an intuitive metric for argument strength.

Consequently, this thesis addresses the question: *How can the strength of arguments be quantified by their distance to the grounded extension using attack deletion?* To answer this, we introduce a new ranking-based semantics. We first introduce a new score for arguments, counting the minimum number of attacks that have to be removed for each argument to be in the grounded extension. Arguments are then ranked on the basis of this score, where a lower score means a higher ranking. This is an intuitive approach to ranking arguments, since it measures arguments by how far they are away from being universally acceptable and can have value in real world applications.

Many real-world argumentation scenarios do not have a clear-cut solution that is acceptable without compromise. Certain risk factors must be willingly accepted for the best possible outcome to occur. The newly introduced ranking-based semantics may have value in finding such compromises.

Example 1 (Potential Use Case). An example of a realistic scenario can show a use for the semantics.

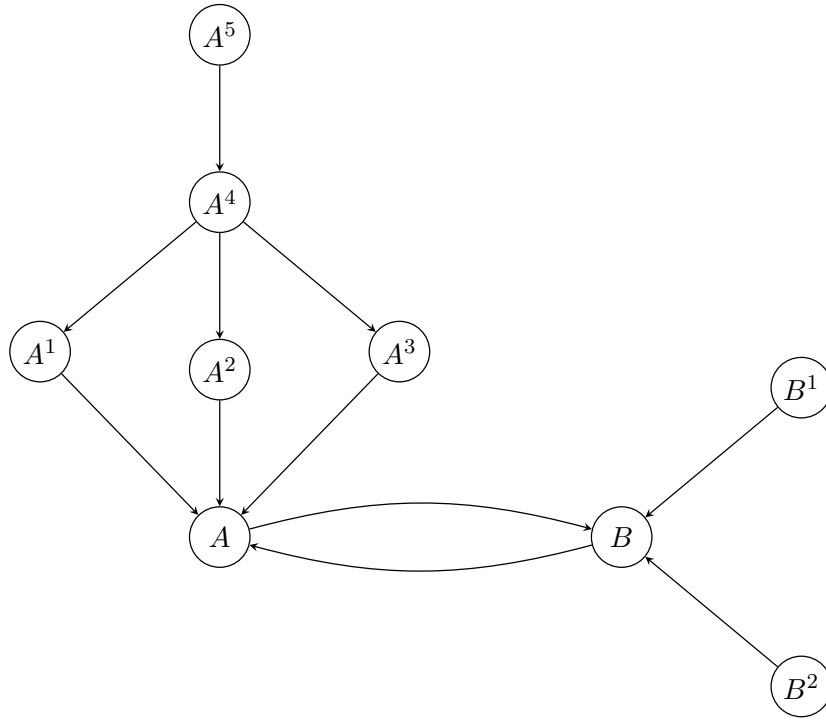


Figure 1: Argumentation framework showing internal attacks on A and B. A's internal defense structure (A^4 countering A^1 – A^3) makes A closer to the grounded extension than B.

The argumentation framework in Figure 1 could show a hypothetical consideration of a family of what kind of car to buy. The choice is between A , an electric car, and B , a gasoline car. The family only wants to buy one car. The following arguments are collected:

- A : The family should buy an electric car.
- A^1 : Electric car's battery degrades over time.
- A^2 : Electric cars are expensive.
- A^3 : Electric cars have little range.
- A^4 : Many conceptions about electric cars are outdated.
- A^5 : The claim from A^4 comes from the brochure of the car manufacturer.
- B : The family should buy a gasoline car.
- B^1 : The running and fuel costs of gasoline cars are high.
- B^2 : Gasoline cars are bad for the environment.

Analyzing arguments A and B using the new ranking-based semantics results in:

$$A \succ_{\mathcal{F}}^{\theta^{AttDel}} B$$

This is because the deletion of the attack (A^5, A^4) would make A part of the grounded extension, while the two attacks from B^1 and B^2 on B would need to be removed to make B be accepted by the grounded extension. Following this result, the family chooses to believe the brochure of the car manufacturer making the argument A^4 part of the grounded extension. This argument defends A from arguments A^1 - A^3 .

We begin by establishing the necessary background on abstract argumentation and existing ranking-based semantics. Building on these foundations, we formally define the new attack deletion scoring function and present an implementation using answer set programming. Subsequently, we derive the ranking-based semantics from this score. Based on an analysis of established principles, we introduce an improved version of the semantics. Finally we introduce an implementation of the new semantics and analyze its computational performance.

2 Background

This chapter outlines the preliminary concepts needed as a basis for the definition, analysis and implementation of the introduced semantics. This includes the general structure of Abstract Argumentation [9], the necessary extension semantics and pre-existing argument-ranking-based semantics. The introduced preliminary concepts are based on work by Dung [9] and Bonzon et al. [6]. Additionally, a declarative method of implementation using answer set programming is introduced.

2.1 Abstract Argumentation

Abstract argumentation was introduced to isolate the mechanism by which humans evaluate competing arguments and to express it in a form suitable for general reasoning [9]. It defines a minimal data structure that abstracts away from internal argument content while retaining the conflict structure that determines acceptability. The outcome is a directed graph in which nodes represent arguments and edges represent attacks, providing an abstract model of the conflict on which semantic analysis can be performed [9].

Definition 1 (Argumentation Framework [9]). An *Argumentation Framework* (AF) $\mathcal{F}$ is a pair:

$$\mathcal{F} = (A, R)$$

where:

- A is a finite set of *arguments*.
- R is a binary relation on A , i.e., $R \subseteq A \times A$. This relation is called the *attack relation*.

For any two arguments $a, b \in A$, the notation $(a, b) \in R$ means that argument a attacks argument b . Intuitively, the attack relation maps the invalidation of argument b by argument a .

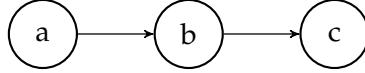


Figure 2: A simple abstract argumentation framework.

Figure 2 shows a basic abstract argumentation framework with three arguments and two attacks with argument a attacking argument b and argument b attacking argument c . Since b is attacked by a , a defends c from the attack by b .

Definition 2 (Defense [1]). A set of arguments $S \subseteq A$ *defends* an argument a , if for all attackers b with $(b, a) \in R$, some argument $c \in S$ attacks b .

Some definitions in this thesis rely on a specific notation for layered sets of attacks and defenses. For an argument $a \in A$, the index $n \in \mathbb{N}_{>0}$ explicitly denotes the length of a directed path to a . Odd path lengths form sets of attackers, denoted $R_n^-(a)$, while even path lengths form sets of defenders, denoted $R_n^+(a)$. Therefore, $R_1^-(a)$ contains all direct attackers of a (equivalent to $\{a\}^-$), $R_2^+(a)$ contains the direct defenders (attackers of $R_1^-(a)$), $R_3^-(a)$ contains the attackers of those defenders, and so on.

2.2 Extension-Based Semantics

Acceptability of arguments is determined by grouping them into extensions, which are subsets of the arguments in the argumentation framework that satisfy certain properties. These properties are solely built upon the previously introduced concepts of attacks and defense. The grounded extension is essential for this thesis. It relates to other extensions which are briefly introduced in the following definitions.

Definition 3 (Conflict-Free Set [9]). A set of arguments $S \subseteq A$ is *conflict-free* if there are no arguments $a, b \in S$ such that $(a, b) \in R$.

Definition 4 (Admissible Extension). A conflict-free set $S \subseteq A$ is *admissible* if every argument in S is defended by S .

Definition 5 (Complete Extension). An admissible set $S \subseteq A$ is a *complete extension* if it contains all arguments it defends.

Definition 6 (Grounded Extension). The *grounded extension* is the least (with respect to set inclusion $\subseteq$) complete extension.

The grounded extension represents the set of arguments that are uncontroversially acceptable. All arguments that are present in every complete extension are in the grounded extension. The formal definition describes the grounded extension as the least fixed point of the characteristic function of the AF.

The *characteristic function* $\Delta_{\mathcal{F}} : 2^A \rightarrow 2^A$ of an AF $\mathcal{F} = (A, R)$ is defined for any set $S \subseteq A$ as:

$$\Delta_{\mathcal{F}}(S) = \{a \in A \mid a \text{ is defended by } S\}$$

The *grounded extension* is the least fixed point of $\Delta_{\mathcal{F}}$:

$$GE = \bigcup_{i \geq 0} \Delta_{\mathcal{F}}^i(\emptyset)$$

This formal definition offers a compact iterative approach of computing the grounded extension by building upon itself, which is important for the context of this thesis. Throughout this thesis, a *grounded argument* refers to an argument within the grounded extension.

Definition 7 (Preferred Extension). The *preferred extension* is the maximal (with respect to set inclusion $\subseteq$) admissible set.

The preferred extension is the largest complete extension of an AF.

Definition 8 (Stable Extension). A conflict-free set $S \subseteq A$ is a *stable extension* if for every argument $a \in A \setminus S$, there exists an argument $b \in S$ such that $(b, a) \in R$.

2.3 Ranking-Based Semantics

Evaluating the acceptability of arguments is a main goal of discussions. To compare arguments on their validity, they can be ranked from strongest to least strong. In abstract argumentation, given the simple structure and the lack of further information, this must be done based solely on the attack relationships between arguments. In contrast to extension-based semantics which provide a binary classification of arguments (accepted or rejected), ranking-based semantics aim to provide a more detailed evaluation by inducing an ordering over the set of arguments.

Definition 9 (Ranking-Based Semantics). A *ranking-based semantics* θ is a function that maps any argumentation framework $\mathcal{F} = (A, R)$ to a ranking $\succeq_{\mathcal{F}}^{\theta}$ on A . This ranking is a preorder (a transitive, reflexive relation).

For any two arguments $a, b \in A$:

- $a \succeq_{\mathcal{F}}^{\theta} b$ means that a is at least as strong as b .
- $a \succ_{\mathcal{F}}^{\theta} b$ (a is strictly stronger than b) holds if $a \succeq_{\mathcal{F}}^{\theta} b$ and not $b \succeq_{\mathcal{F}}^{\theta} a$.

- $a \simeq_{\mathcal{F}}^{\theta} b$ (a and b are equally strong) holds if $a \succ_{\mathcal{F}}^{\theta} b$ and $b \succ_{\mathcal{F}}^{\theta} a$.
- $a \succ_{\mathcal{F}}^{\theta} b$ (a and b are incomparable) holds if no ranking between a and b can be established.

Example 2 (Attack-Based Ranking). The score $Score_{\mathcal{F}}(a)$ for an argument a in an AF $\mathcal{F}$ is the number of its direct attackers:

$$Score_{\mathcal{F}}(a) = |\{a\}_{\mathcal{F}}^{-}|$$

The ranking $\succeq_{\mathcal{F}}^{Att}$ is then defined for any two arguments $a, b \in A$ as:

$$a \succeq_{\mathcal{F}}^{Att} b \iff Score_{\mathcal{F}}(a) \leq Score_{\mathcal{F}}(b)$$

While the ranking $\succeq_{\mathcal{F}}^{\theta}$ directly compares individual arguments, this can be extended to compare sets of arguments, which is essential for certain principles. A set of arguments S_1 is considered at least as acceptable as a set S_2 if there exists an injective function $f : S_2 \rightarrow S_1$ such that for every argument $a \in S_2$, $f(a) \succeq_{\mathcal{F}}^{\theta} a$ [1]. Similarly, strict preference ($S_1 \succ_{\mathcal{F}}^{\theta} S_2$) and equivalence ($S_1 \simeq_{\mathcal{F}}^{\theta} S_2$) between sets can be derived from this [1]. To evaluate ranking-based semantics, principles have been introduced. Based on the compliance of these principles, one can compare different semantics to fit the use case. For this, the concepts of isomorphism and connected components are used [6].

Definition 10 (Isomorphism [6]). Two argumentation frameworks $\mathcal{F} = (A, R)$ and $\mathcal{F}' = (A', R')$ are *isomorphic* if there is a bijective function $\rho : A \rightarrow A'$ such that for all $a, b \in A$, we have $(a, b) \in R \iff (\rho(a), \rho(b)) \in R'$, i.e., the AF graph structures are identical.

Definition 11 (Connected Component [6]). A *connected component* of an AF $\mathcal{F}$ ($CC(\mathcal{F})$) is a maximal subgraph where for any two arguments a, b in $CC(\mathcal{F})$, there exists a path between a and b , ignoring the direction of the attacks.

2.4 Implementation Preliminaries

The implementation chapter will show that implementations of the new semantics will run into performance challenges, as the problem is complex [18]. Declarative approaches which are highly optimized for solving optimization problems are able to handle large frameworks [18]. Answer set programming is one of these declarative solving approaches, and will be used for the implementation in this thesis. Though it is not the fastest approach for the issue at hand [18], it is relatively easy to work with and gives a first insight into, and helps with the analysis of the semantics. A more efficient approach can be used later on [18]. Answer Set Programming (ASP) is a declarative programming paradigm for solving complex search problems [14]. Given a problem encoded as rules and constraints, it derives admissible answer sets by logical inference over the provided information. Solving is performed by Clingo,

a grounder–solver system that converts a logic program into a propositional representation and computes its stable models via conflict-driven search [15]. An ASP program consists of facts, choice rules, normal rules, constraints, and directives [14].

Example 3 (ASP Example). Using a simple example, these components of the program are explained.

```

person(alice).                                     (1)
person(bob).
person(carol).
friends(alice, bob).
friends(bob, carol).
friends(carol, alice).
friends(bob, alice).

{ likes_most(X,Y) : friends(X,Y) }.                (2)

best_friends(X,Y) :- likes_most(X,Y), likes_most(Y,X). (3)

:- likes_most(X,Y), likes_most(X,Z), Y != Z.       (4)
:- person(X), not likes_most(X,_).                 (5)

#maximize{1,X,Y : best_friends(X,Y)}.               (6)
#show likes_most/2.                                 (7)
#show best_friends/2.                               (8)

```

Facts (1) describe indisputable atoms that are unchanged. The selection that creates satisfactory set models is defined by *Choices*. The Choice in this example (2) specifies, that a set of `likes_most` atoms is selected for the program to be satisfiable. *Rules* define relations between atoms. The rule in this example (3) specifies, that for two friends to be best friends, they must like each other the most. *Constraints* specify things that should not happen. In this case, it is defined, that a person can only like one other person the most (4) and that every person must like some person the most (5). The directives take this logical program and define what to solve for. In this case, the solver is assigned to find the model with the largest number of `best_friends` (6) and to return the corresponding `likes_most` (7) and `best_friends` (8) relations. The Friend relation is visualized in Figure 3.

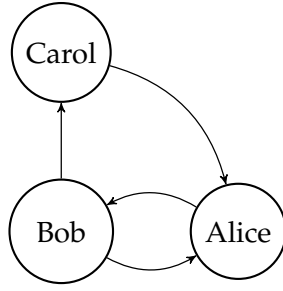
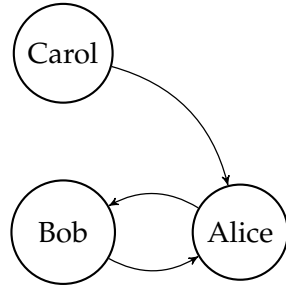
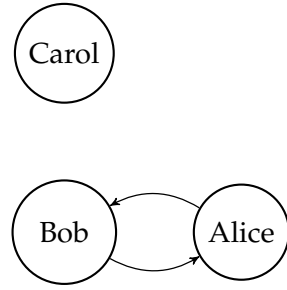


Figure 3: Friend relation from Example 3.

Solving for maximum `best_friends` relations, the resulting answer set is visualized in Figure 4:



(a) Likes Most Relations



(b) Best Friends Relations

Figure 4: Friend Relations based on Clingo Output

Modeling problems in logic programs and solving using Clingo is a powerful tool that can solve a wide range problems [14]. It can be used for finding optimal solutions or all possible solutions efficiently [14].

3 Attack Deletion Ranking-Based Semantics

Modifying the structure of an argumentation framework to change argument acceptability is an established concept in the literature known as *extension enforcement* [18]. Traditionally, extension enforcement explores how operations like adding or deleting attacks can force a specific argument or set of arguments to become accepted under a given semantics [8]. We adapt this concept from a modification tool into an evaluation metric. This section introduces a scoring function that counts the minimal number of attack deletions required to enforce a given argument's inclusion in the grounded extension. Building from this, a ranking-based semantics is introduced, which ranks arguments based solely on their score.

3.1 Attack Deletion Scoring Function

The score reflects how resilient an argument is: arguments that require many attack deletions are far from being universally acceptable and receive a high score, while arguments that are in the grounded extension receive a score of 0, as no attack deletions are required to include them in the grounded extension.

Definition 12 (Attack Deletion Optimization for Grounded Extension ($AttDelScore$)¹). Given an AF $\mathcal{F} = (A, R)$ with an argument $a \in A$ and a predicate

$$Gr(\mathcal{F}', a) = \{a \text{ is in the grounded extension of } \mathcal{F}'\}$$

find a minimal k for which a subset $R' \subseteq R$ exists such that $|R'| = k$, $\mathcal{F}' = (A, R \setminus R')$ and $Gr(\mathcal{F}', a)$ is true.

As this problem is a version of *non-strict extension enforcement under grounded semantics* with a single element input set, the underlying decision problem is NP-complete [19]. Consequently, the associated optimization problem, minimizing the number of attack deletions to achieve this enforcement, is inherently NP-hard [13, 18].

3.2 Attack Deletion Ranking-Based Semantics

Building upon the attack deletion score, we define a new ranking-based semantics that orders arguments by their distance to the grounded extension. Arguments requiring fewer attack deletions to enter the grounded extension are ranked higher, reflecting their closeness to universal acceptability. This provides an intuitive and interpretable measure of argument strength. Formally, the ranking is defined using the minimal attack deletion score as the basis for comparison.

¹The definition is based on the definition of grounded extension enforcement by Niskanen et al. [18].

Definition 13 (Attack Deletion Grounded Extension Enforcement Ranking-Based Semantics (θ^{AttDel})). Given an AF $\mathcal{F} = (A, R)$, define a ranking-based semantics θ^{AttDel} . For all $a, b \in A$:

$$a \succeq_{\mathcal{F}}^{\theta^{AttDel}} b \iff AttDelScore_{\mathcal{F}}(a) \leq AttDelScore_{\mathcal{F}}(b)$$

Based on the weak order $\succeq_{\mathcal{F}}^{\theta^{AttDel}}$, the strict ranking and equivalence relations are naturally derived as follows:

- $a \succ_{\mathcal{F}}^{\theta^{AttDel}} b$ iff $a \succeq_{\mathcal{F}}^{\theta^{AttDel}} b$ and $b \not\succeq_{\mathcal{F}}^{\theta^{AttDel}} a$, which is the case if $AttDelScore_{\mathcal{F}}(a) < AttDelScore_{\mathcal{F}}(b)$.
- $a \simeq_{\mathcal{F}}^{\theta^{AttDel}} b$ iff $a \succeq_{\mathcal{F}}^{\theta^{AttDel}} b$ and $b \succeq_{\mathcal{F}}^{\theta^{AttDel}} a$, which is the case if $AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(b)$.

Due to the nature of $AttDelScore_{\mathcal{F}}$ mapping all arguments to non-negative integers, the relation $\succeq_{\mathcal{F}}^{\theta^{AttDel}}$ is inherently a total and transitive relation, fulfilling the criteria for a ranking, when applied to an AF $\mathcal{F}$ [1]. Throughout this thesis, θ^{AttDel} is regularly referred to as AttDel.

To establish an intuition for this ranking system compared to traditional extension semantics, we consider the distinction between arguments that extension semantics would evaluate equally.

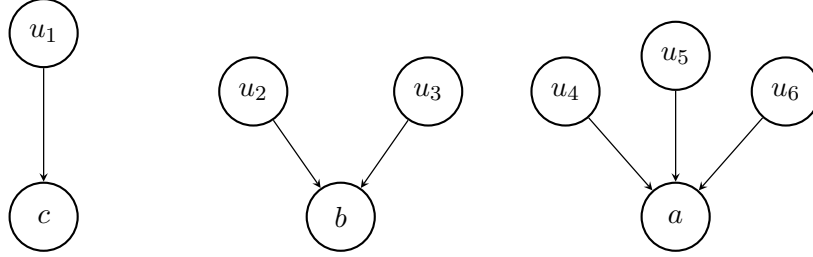


Figure 5: AF $\mathcal{F}$ demonstrating graduality in the AttDel semantics compared to the grounded extension.

In $\mathcal{F}$ from Figure 5, the unattacked arguments $u_1 - u_6$ are part of the grounded extension, with a, b and c being rejected. AttDel ranks the arguments $u_1 - u_6$, which are unattacked and therefore have an $AttDelScore_{\mathcal{F}}$ of 0, highest. Because the deletion of their direct attackers is required, the scores of a, b and c are 3, 2 and 1 respectively, leading to the intuitive sequence $c \succ_{\mathcal{F}}^{AttDel} b \succ_{\mathcal{F}}^{AttDel} a$. This yields the complete equivalence class and ranking sequence:

$$u_1 \simeq_{\mathcal{F}}^{AttDel} u_2 \simeq_{\mathcal{F}}^{AttDel} u_3 \simeq_{\mathcal{F}}^{AttDel} u_4 \simeq_{\mathcal{F}}^{AttDel} u_5 \simeq_{\mathcal{F}}^{AttDel} u_6 \succ_{\mathcal{F}}^{AttDel} c \succ_{\mathcal{F}}^{AttDel} b \succ_{\mathcal{F}}^{AttDel} a$$

We find a clear coherency between the grounded extension and AttDel, as all arguments accepted by the grounded extension are ranked highest by AttDel and all

arguments rejected by the grounded extension rank lower than these accepted ones. AttDel therefore serves as a bridge between the simple and intuitive extension semantics and the more complex ranking-based semantics.

4 Principle Analysis

Having formally defined the Attack Deletion ranking-based semantics (θ^{AttDel}), it is essential to evaluate its theoretical behavior against the standard principles established in the literature [6]. This axiomatic analysis serves to benchmark the new semantics against existing approaches, potentially revealing a novel behavioral profile and also finding structural weaknesses of the AttDel semantics compared to other benchmarked semantics. The aim is to characterize the semantics based on its structural properties, determining specifically how it prioritizes conflicting intuitions, such as the trade-off between the quantity and the quality of attackers.

4.1 Abstraction

Principle 1 (Abstraction (Abs) [6]). A ranking-based semantics θ satisfies *Abstraction* if for any two isomorphic AFs $\mathcal{F}$ and $\mathcal{F}'$, the following holds for all $a, b \in A$:

$$a \succeq_{\mathcal{F}}^{\theta} b \iff \rho(a) \succeq_{\mathcal{F}'}^{\theta} \rho(b)$$

Intuitively, this means that any renaming of arguments will not change the ranking between them.

Proposition 1 (Abstraction).

The AttDel semantics complies with abstraction.

Proof. For our semantics, this holds if for any isomorphic AFs $\mathcal{F} = (A, R)$ and $\mathcal{F}' = (A', R')$ mapped by an isomorphism γ , for any $a \in A$ the following holds:

$$AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}'}(\gamma(a))$$

Let $\mathcal{F} = (A, R)$ and $\mathcal{F}' = (A', R') = (\gamma(A), \gamma(R))$ be isomorphic under γ . We prove that for any $x \in A$ and its image $x' = \gamma(x) \in A'$, it holds that $AttDelScore_{\mathcal{F}}(x) = AttDelScore_{\mathcal{F}'}(x')$.

Given $AttDelScore_{\mathcal{F}}(x) = k$, there exists a set $R^{mod} \subseteq R$ where $|R^{mod}| = k$ such that x is in the grounded extension of $\mathcal{F}^{mod} = (A, R \setminus R^{mod})$. Since the grounded extension satisfies the abstraction principle as it is solely defined by the graph structure [9], if $\sigma_{\mathcal{F}}^{gr}$ is the grounded extension of $\mathcal{F}$, then $\gamma(\sigma_{\mathcal{F}}^{gr})$ is the grounded extension of $\mathcal{F}'$.

The isomorphic graph structure implies that x' is in the grounded extension of $\mathcal{F}'^{mod} = (A', R' \setminus \gamma(R^{mod}))$. Since isomorphism is a bijective relation, $|R^{mod}| = |\gamma(R^{mod})|$. Furthermore, since γ is reversible, the removal of no smaller set than $|\gamma(R^{mod})|$ can enforce x' in the grounded extension of $\mathcal{F}'$, as its inverse would cause

a smaller set than R' to enforce the grounded extension for x , contradicting minimality.

Consequently, $AttDelScore_{\mathcal{F}}(x) = AttDelScore_{\mathcal{F}'}(x')$. Because the scores are identical across the isomorphism, the ranking defined by the semantics is preserved. Thus, the AttDel semantics complies with abstraction. $\square$

4.2 Independence

Principle 2 (Independence [6]). A ranking-based semantics complies with *Independence*, if for any AF $\mathcal{F} = (A, R)$ where

$$A' \in CC(\mathcal{F}), a, b \in A', a \succeq_{\mathcal{F}}^{\theta} b \text{ iff } a \succeq_{\mathcal{F} \downarrow A'}^{\theta} b$$

If one argument ranks higher than another argument within the ranking semantics of a connected component, it also ranks higher within the whole AF.

Proposition 2 (Independence).

The AttDel semantics complies with independence.

Proof. For our semantics, given any AF $\mathcal{F} = (A, R)$ and any connected component $\mathcal{F}' = (A', R') \in CC(\mathcal{F})$, this is satisfied if for all $a \in A'$:

$$AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}'}(a)$$

Let $\mathcal{F} = (A, R)$ be an AF, $\mathcal{F}' = (A', R') \in CC(\mathcal{F})$ be a connected component, and $a \in A'$ be an argument. We must prove that the minimal set of attack deletions R'^{-} required to enforce a in the grounded extension of $\mathcal{F}'$ is identical to the minimal set R^{-} required to enforce a in the grounded extension of $\mathcal{F}$.

By the directionality principle of the grounded extension [3], the acceptability of an argument depends exclusively on the acceptability of its ancestors. By definition of a connected component [6], there exists no attack $r = (x, y) \in R$ such that $x \notin A'$ and $y \in A'$. Therefore, the grounded acceptance of a is completely isolated from any graph structure outside of $\mathcal{F}'$.

Consequently, any set of attack deletions R'^{-} that places a in the grounded extension of the component $\mathcal{F}'^{mod} = (A', R' \setminus R'^{-})$ inherently places a in the grounded extension of the full framework $\mathcal{F}^{mod} = (A, R \setminus R'^{-})$.

To prove minimality, assume there exists a set $R^{-} \subseteq R$ such that $|R^{-}| < |R'^{-}|$ and a is in the grounded extension of $(A, R \setminus R^{-})$. Due to the directionality principle, any deletion in R^{-} involving arguments outside of A' is strictly redundant for the acceptance of a . This implies that a purely internal subset exists that is smaller than R'^{-} , which directly contradicts the premise that R'^{-} is minimal for $\mathcal{F}'$.

Consequently, $AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}'}(a)$. Because the score of any argument is identical whether evaluated globally or within its isolated component, the internal ranking is preserved globally.

Thus, the AttDel semantics complies with independence. $\square$

4.3 Total

Principle 3 (Total (Tot) [6]). A ranking-based semantics θ complies with the *Total* principle, if for any two arguments $a, b \in A$ in an AF $\mathcal{F} = (A, R)$:

$$a \succeq_{\mathcal{F}}^{\theta} b \quad \text{or} \quad b \succeq_{\mathcal{F}}^{\theta} a$$

Any two arguments can be ranked against each other.

Proposition 3 (Total).

The AttDel semantics complies with total.

Proof. For our semantics, since the ranking is determined by $a \succeq_{\mathcal{F}}^{\theta^{AttDel}} b \iff AttDelScore_{\mathcal{F}}(a) \leq AttDelScore_{\mathcal{F}}(b)$, this is satisfied if for any argument $a \in A$:

$$AttDelScore_{\mathcal{F}}(a) \in \mathbb{N}$$

Let $\mathcal{F} = (A, R)$ be an AF, and let $a, b \in A$. We must prove that there are no incomparable arguments, meaning their respective scores can always be mathematically compared.

By definition, the $AttDelScore_{\mathcal{F}}(a)$ corresponds to the cardinality of a minimal subset of attacks $R' \subseteq R$ whose deletion enforces a in the grounded extension. R' always exists, because if all direct attackers of a are deleted, a is inherently part of the grounded extension. Therefore: $|R'| \leq |\{a\}^-|$. Since the AF is finite [9], the set of attacks R is finite. The cardinality of any subset of a finite set is strictly a non-negative integer.

Consequently, $AttDelScore_{\mathcal{F}}(a)$ and $AttDelScore_{\mathcal{F}}(b)$ are both integers. Due to the total order properties of integers, it must unconditionally hold that either

$$AttDelScore_{\mathcal{F}}(a) \leq AttDelScore_{\mathcal{F}}(b)$$

or

$$AttDelScore_{\mathcal{F}}(b) \leq AttDelScore_{\mathcal{F}}(a)$$

implying $a \succeq_{\mathcal{F}}^{AttDel} b$ or $b \succeq_{\mathcal{F}}^{AttDel} a$.

Thus, the AttDel semantics complies with total. $\square$

4.4 Non-Attacked Equivalence

Principle 4 (Non-Attacked Equivalence (NaE) [6]). A ranking-based semantics complies with this principle, if all non-attacked arguments are ranked the same. Formally, under a ranking-based semantics θ , if for an AF $\mathcal{F} = (A, R)$ with any arguments $a, b \in A$, where $R_1^-(a) = R_1^-(b) = \emptyset$:

$$a \simeq_{\mathcal{F}}^{\theta} b$$

Proposition 4 (Non-Attacked Equivalence).

The AttDel semantics complies with non-attacked equivalence.

Proof. For our semantics, given any AF $\mathcal{F} = (A, R)$ and any unattacked arguments $a, b \in A$ where $R_1^-(a) = R_1^-(b) = \emptyset$, this is satisfied if:

$$AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(b)$$

Let $\mathcal{F} = (A, R)$ be an AF, and let $a, b \in A$ be unattacked arguments. We prove this by showing that for any unattacked argument, the $AttDelScore$ is 0.

By the definition of the grounded extension, any argument with no attackers is automatically included in the grounded extension. Therefore, the minimal set of attacks required to be removed for a (and b) to be in the grounded extension is the empty set $S = \emptyset$. Since $|S| = 0$, every unattacked argument has a score of 0.

Consequently, $AttDelScore_{\mathcal{F}}(a) = 0$ and $AttDelScore_{\mathcal{F}}(b) = 0$, meaning $AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(b)$. This implies $a \simeq_{\mathcal{F}}^{AttDel} b$.

Thus, the AttDel semantics complies with non-attacked equivalence. $\square$

4.5 Void Precedence

Principle 5 (Void Precedence (VP) [6]). A ranking-based semantics complies with *Void Precedence*, if for any AF $\mathcal{F} = (A, R)$:

$$a, b \in A, \{a\}_{\mathcal{F}}^- = \emptyset \text{ and } \{b\}_{\mathcal{F}}^- \neq \emptyset \text{ then } a \succ_{\mathcal{F}}^{\theta} b$$

Intuitively, if any unattacked argument ranks strictly higher than any attacked argument, a ranking-based semantics θ complies with void precedence.

Proposition 5 (Void Precedence).

The AttDel semantics does not comply with void precedence.

Proof. This is demonstrated by the counter-example in Figure 6.

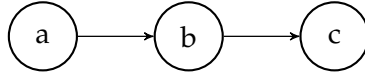


Figure 6: Counter-example AF $\mathcal{F}$ for void precedence.

In $\mathcal{F}$, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a) = 0$ (because it is unattacked and part of the grounded extension).
- $AttDelScore_{\mathcal{F}}(c) = 0$ (because it is defended by a , making it part of the grounded extension).

Based on these scores, $AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(c)$, which yields $a \simeq_{\mathcal{F}}^{AttDel} c$. However, under void precedence, because a is unattacked ($\{a\}_{\mathcal{F}}^- = \emptyset$) and c is attacked ($\{c\}_{\mathcal{F}}^- = \{b\}$), the strict ranking $a \succ_{\mathcal{F}}^{\theta} c$ is required.

This contradiction proves that the AttDel semantics does not comply with void precedence. $\square$

4.6 Cardinality Precedence

Principle 6 (Cardinality Precedence (CP) [6]). For an argument a in an AF $\mathcal{F}$, the set $R_1^-(a)$ is the set of all arguments that attack a . A ranking-based semantics θ complies with *Cardinality Precedence*, if for all arguments $a, b \in A$ in an AF $\mathcal{F}$:

$$|R_1^-(a)| < |R_1^-(b)| \text{ then } a \succ_{\mathcal{F}}^{\theta} b$$

Under a ranking-based semantics complying with cardinality precedence, an argument with fewer attackers than another will rank higher than ones with more attackers.

Proposition 6 (Cardinality Precedence).

The AttDel semantics does not comply with cardinality precedence.

Proof. This is demonstrated by the counter-example in Figure 6 from the previous section.

In the AF $\mathcal{F}$ depicted in Figure 6, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a) = 0$
- $AttDelScore_{\mathcal{F}}(c) = 0$

Based on these scores, $AttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(c)$, which yields $a \simeq_{\mathcal{F}}^{AttDel} c$. However, under cardinality precedence, because a has an empty set of direct attackers ($|R_1^-(a)| = 0$) and c has a non-empty set of direct attackers ($|R_1^-(c)| = 1$), the strict ranking $a \succ_{\mathcal{F}}^{\theta} c$ is required.

This contradiction proves that the AttDel semantics does not comply with cardinality precedence. $\square$

4.7 Quality Precedence

Principle 7 (Quality Precedence (QP) [6]). The *Quality Precedence* principle checks the quality of a single attacking argument. A ranking-based semantics θ complies with quality precedence if for any AF $\mathcal{F} = (A, R)$ with arguments $a, b \in A$ if there exists an argument $b' \in \{b\}^-$ for which any argument $a' \in \{a\}^-$:

$$b' \succ_{\mathcal{F}}^{\theta} a' \text{ then } a \succ_{\mathcal{F}}^{\theta} b$$

Intuitively, if an argument b has an attacker strictly stronger than all attackers of argument a , then a is strictly stronger than b .

Proposition 7 (Quality Precedence).

The AttDel semantics does not comply with quality precedence.

Proof. This is demonstrated by the counter-example in Figure 7.

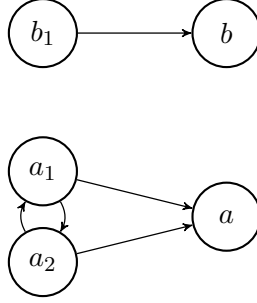


Figure 7: Counter-example AF $\mathcal{F}$ for quality precedence.

In the AF $\mathcal{F}$ depicted in Figure 7, there are five arguments for which the attack deletion score can be calculated:

- $AttDelScore_{\mathcal{F}}(a_1) = 1$
- $AttDelScore_{\mathcal{F}}(a_2) = 1$
- $AttDelScore_{\mathcal{F}}(a) = 2$
- $AttDelScore_{\mathcal{F}}(b_1) = 0$
- $AttDelScore_{\mathcal{F}}(b) = 1$

By these scores, we can deduce $b \succ_{\mathcal{F}}^{AttDel} a$. Under quality precedence, this would imply that the strongest attacker of b is weaker than some attacker of a . This, however, is not the case, as both arguments in $\{a\}^-$ have a score of 1, while the score of the only attacker of b , b_1 has a score of 0. The strongest attacker of b is therefore stronger than the strongest attacker of a .

This contradiction proves that the AttDel semantics does not comply with quality precedence. $\square$

4.8 Self-Contradiction

Principle 8 (Self-Contradiction (SC) [6]). A ranking-based semantics θ complies with *Self-Contradiction*, if for any AF $\mathcal{F} = (A, R)$:

$$a, b \in A, \text{ if } (a, a) \notin R \text{ and } (b, b) \in R, \text{ then } a \succ_{\mathcal{F}}^{\theta} b$$

Intuitively, self attacking arguments are always strictly ranked lower than non-self attacking arguments.

Proposition 8 (Self-Contradiction).

The AttDel semantics does not comply with self-contradiction.

Proof. This is demonstrated by the counter-example in Figure 8.

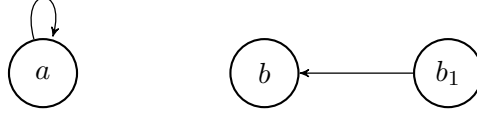


Figure 8: Counter-example AF $\mathcal{F}$ for self-contradiction.

In the AF $\mathcal{F}$ depicted in Figure 8, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a) = 1$
- $AttDelScore_{\mathcal{F}}(b) = 1$

In the AF $\mathcal{F}$, a is self-attacking and, under self-contradiction, $b \succ_{\mathcal{F}}^{AttDel} a$ must hold. However, $AttDelScore_{\mathcal{F}}(b) = AttDelScore_{\mathcal{F}}(a) = 1$ and therefore $a \simeq_{\mathcal{F}}^{AttDel} b$. This contradiction proves that the AttDel semantics does not comply with self-contradiction. $\square$

4.9 Counter-Transitivity

Principle 9 (Counter-Transitivity (CT) [6]). A ranking-based semantics θ complies with *Counter-Transitivity* if for all AF $\mathcal{F} = (A, R)$:

$$a, b \in A, \{b\}_{\mathcal{F}}^- \succeq_{\mathcal{F}}^{\theta} \{a\}_{\mathcal{F}}^- \text{ then } a \succeq_{\mathcal{F}}^{\theta} b$$

Intuitively, a is at least as strong as b if all attackers of b are at least as strong as all attackers of a .

Proposition 9 (Counter-Transitivity).

The AttDel semantics does not comply with counter-transitivity.

Proof. This is demonstrated by the counter-example in Figure 9.

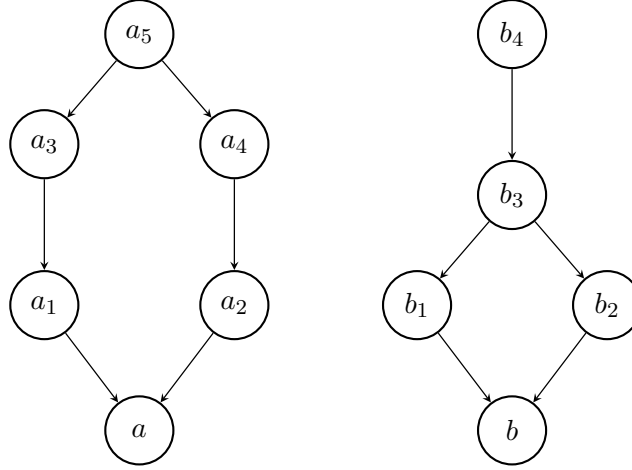


Figure 9: Counter-example AF $\mathcal{F}$ for counter-transitivity.

In the AF $\mathcal{F}$ depicted in Figure 9, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a_1) = 0$
- $AttDelScore_{\mathcal{F}}(a_2) = 0$
- $AttDelScore_{\mathcal{F}}(b_1) = 0$
- $AttDelScore_{\mathcal{F}}(b_2) = 0$
- $AttDelScore_{\mathcal{F}}(a) = 2$
- $AttDelScore_{\mathcal{F}}(b) = 1$

For each of the direct attackers of a , a_1 and a_2 ,

$$AttDelScore_{\mathcal{F}}(a_1) = AttDelScore_{\mathcal{F}}(a_2) = 0$$

the same holds for the direct attackers of b , b_1 and b_2 . Under this condition, when complying with counter-transitivity, $a \simeq_{\mathcal{F}}^{AttDel} b$ must hold due to the symmetrical nature of the principle. If a ranking-based semantics θ ranks the attackers of x and y as equally strong, then for θ to satisfy counter-transitivity, $x \simeq^{\theta} y$ must hold. Since the principle requires $x \succeq^{\theta} y$ when y 's attackers are at least as strong as x 's, the symmetrical nature of equal attackers necessitates both $x \succeq^{\theta} y$ and $y \succeq^{\theta} x$, forcing an equivalence.

In the AF $\mathcal{F}$ depicted in Figure 9, $AttDelScore_{\mathcal{F}}(a) = 2$, since deleting the two direct attacks of a would cause a to be unattacked and part of the grounded extension. $AttDelScore_{\mathcal{F}}(b) = 1$, since deleting the attack from b_4 to b_3 causes b_3 to defend b from its direct attackers. $b \succ_{\mathcal{F}}^{AttDel} a$ holds, contradicting $a \simeq_{\mathcal{F}}^{AttDel} b$.

This contradiction proves that the AttDel semantics does not comply with counter-transitivity. $\square$

This example shows the valuation of the new semantics of bottlenecks in an argumentation structure. Even if the attackers of a and b seem identically strong, a 's attackers build upon a more structurally sound basis of argumentation, requiring more compromises to accept a , than to accept b .

Principle 10 (Strict Counter-Transitivity (SCT) [6]). A ranking-based semantics θ complies with *Strict Counter-Transitivity*, if for all AF $\mathcal{F} = (A, R)$:

$$a, b \in A, \{b\}_{\mathcal{F}}^{-} \succ_{\mathcal{F}}^{\theta} \{a\}_{\mathcal{F}}^{-} \text{ then } a \succ_{\mathcal{F}}^{\theta} b$$

Intuitively, this principle works analogous to counter-transitivity with strict acceptability ranking. a is stronger than b if all attackers of b are stronger than all attackers of a .

Proof. With SCT implying CT [6], because AttDel does not comply with CT, SCT can also not hold for the semantics. $\square$

4.10 σ -Compatibility

Principle 11 (σ -Compatibility [5]). For an extension σ , a ranking-based semantics θ complies with σ -Compatibility, if for all AF $\mathcal{F} = (A, R)$:

$$\text{if } a \in \bigcup \sigma(\mathcal{F}) \text{ and } b \notin \bigcup \sigma(\mathcal{F}) \text{ then } a \succ_{\mathcal{F}}^{\theta} b$$

Intuitively, all arguments that are accepted by the σ -extension are stronger than the arguments outside of that extension.

In the following section, we investigate the compliance of σ -compatibility with respect to the AttDel semantics, going over the common extension-based semantics.

Proposition 10 (*adm*-Compatibility).

The AttDel semantics does not comply with *adm*-compatibility.

Proof. This is demonstrated by the counter-example in Figure 10.

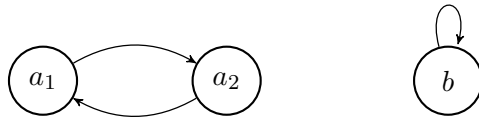


Figure 10: Counter-example AF $\mathcal{F}$ for *adm*-compatibility.

In the AF $\mathcal{F}$ depicted in Figure 10, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a_1) = 1$
- $AttDelScore_{\mathcal{F}}(a_2) = 1$

- $AttDelScore_{\mathcal{F}}(b) = 1$

The admissible sets in this example are $\emptyset$, $\{a_1\}$ and $\{a_2\}$, as a_1 can defend itself against a_2 and vice versa. Note that b is not inside an admissible set, since it attacks itself. Since $AttDelScore_{\mathcal{F}}(a_1) = AttDelScore_{\mathcal{F}}(a_2) = AttDelScore_{\mathcal{F}}(b) = 1$, the deletion of the only attacking argument would make each argument part of the grounded extension, showing $a_1 \simeq_{\mathcal{F}}^{AttDel} b$. Under *adm-compatibility* $a_1 \succ_{\mathcal{F}}^{AttDel} b$ must hold.

This contradiction proves that the AttDel semantics does not comply with *adm-compatibility*. $\square$

Proposition 11 (co-Compatibility).

The AttDel semantics does not comply with *co-compatibility*.

Proof. This is demonstrated by the counter-example AF $\mathcal{F}$ in Figure 10.

Under this example, since the complete extension is a set of arguments that is admissible and contains each argument that it defends, we find that the complete extensions of $\mathcal{F}$ are $\emptyset$, $\{a_1\}$ and $\{a_2\}$. Since b is not accepted by any complete extension, with the *AttDelScore* being equal among all arguments, the resulting equivalence $a_1 \simeq_{\mathcal{F}}^{AttDel} b$ violates the principle.

This contradiction proves that the AttDel semantics does not comply with *co-compatibility*. $\square$

Proposition 12 (pr-Compatibility).

The AttDel semantics does not comply with *pr-compatibility*.

Proof. This is demonstrated by the counter-example AF $\mathcal{F}$ in Figure 10.

Since the preferred extension is the largest possible admissible set, with this being the case for $\{a_1\}$ and $\{a_2\}$ in $\mathcal{F}$, b is non-accepted by any preferred extension. The resulting equivalence $a_1 \simeq_{\mathcal{F}}^{AttDel} b$ contradicts the requirement $a_1 \succ_{\mathcal{F}}^{AttDel} b$.

This contradiction proves that the AttDel semantics does not comply with *pr-compatibility*. $\square$

Proposition 13 (st-Compatibility).

The AttDel semantics does not comply with *st-compatibility*.

Proof. This is demonstrated by the counter-example in Figure 11.

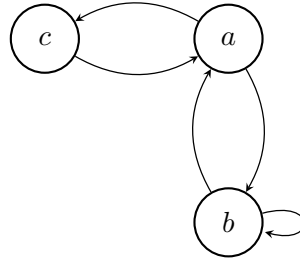


Figure 11: Counter-example AF $\mathcal{F}$ for *st-compatibility*.

In the AF $\mathcal{F}$ depicted in Figure 11, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a) = 2$
- $AttDelScore_{\mathcal{F}}(b) = 2$

In $\mathcal{F}$, argument a attacks all other arguments except itself and $\{a\}$ is the only stable extension of $\mathcal{F}$, because c does not attack b and b attacks itself. Ranking a and b against each other yields $a \simeq_{\mathcal{F}}^{AttDel} b$, which contradicts *st-compatibility*, as a is part of a stable extension, which b is not.

This contradiction proves that the AttDel semantics does not comply with *st-compatibility*. $\square$

Proposition 14 (*gr-Compatibility*).

The AttDel semantics complies with *gr-compatibility*.

Proof. For our semantics, given an AF $\mathcal{F} = (A, R)$ and arguments $a \in GR(\mathcal{F})$ and $b \notin GR(\mathcal{F})$, this is satisfied if:

$$AttDelScore_{\mathcal{F}}(a) < AttDelScore_{\mathcal{F}}(b)$$

Let $\mathcal{F} = (A, R)$ be an AF. We prove this by the definition of the AttDel scores relative to the grounded extension.

Any argument a that is accepted by the grounded extension has an *AttDelScore* of 0, because no attacks need to be removed to include it. Conversely, any argument b that is not accepted by the grounded extension requires the deletion of at least one attack to be included, resulting in an *AttDelScore* of $k \geq 1$. Consequently, arguments accepted by the grounded extension always rank highest and higher than any unaccepted arguments.

Thus, the AttDel semantics complies with *gr-compatibility*. $\square$

4.11 Defense Precedence

Principle 12 (Defense Precedence (DP) [1]). A ranking-based semantics θ complies with *Defense Precedence* if for an AF $\mathcal{F}$ when two arguments $a, b \in A$ have the same number of direct attackers, $|R_1^-(a)| = |R_1^-(b)|$, a is defended and b is not defended, $R_2^+(a) \neq \emptyset$ and $R_2^+(b) = \emptyset$, then $a \succ_{\mathcal{F}}^{\theta} b$. Intuitively, defense precedence only takes two surrounding layers of attacks on an argument into account: the direct attackers and the first-layer defenders.

Proposition 15 (Defense Precedence).

The AttDel semantics does not comply with defense precedence.

Proof. This is demonstrated by the counter-example in Figure 12.

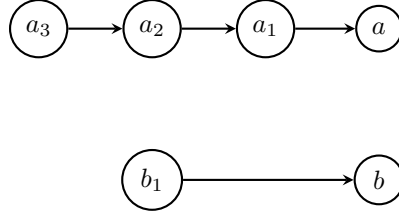


Figure 12: Counter-example AF $\mathcal{F}$ for defense precedence.

In the AF $\mathcal{F}$ depicted in Figure 12, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a) = 1$
- $AttDelScore_{\mathcal{F}}(b) = 1$

If in an AF $\mathcal{F} = (A, R)$ with $a, a_2 \in A$ when $a_2 \in R_2^+(a)$, a is defended by a_2 , however, if a_2 is attacked by an unattacked argument, an attack must be removed for a to be defended by the grounded extension, resulting in $AttDelScore_{\mathcal{F}}(a) = 1$, which is equal to $AttDelScore_{\mathcal{F}}(b)$. Even though a is defended and b is not, $a \simeq_{\mathcal{F}}^{AttDel} b$ follows.

This contradiction proves that the AttDel semantics does not comply with defense precedence. $\square$

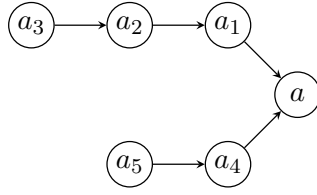
4.12 Distributed-Defense Precedence

Principle 13 (Distributed-Defense Precedence (DDP) [1]). A defense of an argument is *simple*, if every defender of that argument attacks exactly one direct attacker of that argument. A defense of an argument is *distributed*, if every direct attacker is attacked by no more than one defender. A ranking-based semantics θ complies with *Distributed-Defense Precedence*, if for two arguments $a, b \in A$ in an AF $\mathcal{F}$ with $|R_1^-(a)| = |R_1^-(b)|$, with the defense of both being simple, but the defense of a is distributed and the defense of b is not, then $a \succ_{\mathcal{F}}^{\theta} b$. Intuitively, distributed-defense precedence suggests that spreading defenses across different attackers is stronger than concentrating them.

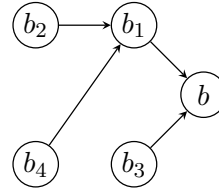
Proposition 16 (Distributed-Defense Precedence).

The AttDel semantics does not comply with distributed-defense precedence.

Proof. This is demonstrated by the counter-example in Figure 13.



Distributed defense of a



Concentrated defense of b

Figure 13: Counter-example AF $\mathcal{F}$ for distributed-defense precedence.

In the AF $\mathcal{F}$ depicted in Figure 13, the relevant arguments evaluate to the following attack deletion scores:

- $AttDelScore_{\mathcal{F}}(a) = 1$
- $AttDelScore_{\mathcal{F}}(b) = 1$

While the formal requirements for distributed-defense precedence are obeyed by a and b , with both having two attackers, two defenders, their defenses being simple and the defense of a being distributed, with the defenders injectively attacking attackers only once, a does not rank higher than b . a_3 is an argument part of the grounded extension attacking the defender a_2 . For the grounded extension to defend a , one of the attacks on this branch must be removed leading to $AttDelScore_{\mathcal{F}}(a) = 1$. b being undefended from b_3 requires the attack from b_3 to be removed causing $AttDelScore_{\mathcal{F}}(b) = 1$. $a \simeq_{\mathcal{F}}^{AttDel} b$ holds.

This contradiction proves that the AttDel semantics does not comply with distributed-defense precedence. $\square$

4.13 Change-Related Principles

The following principles require alternative AFs to compare the same argument a in unequal conditions. Specifically, the alternative graph $\mathcal{F}'$ is a clone of the existing graph $\mathcal{F}$ adding additional branches to the argument a . These branches consist of a non-empty set of arguments which are linearly connected by attacks ending at a . An attack branch br^{Att} is such a branch of odd length, while a defense branch br^{Def} is such a branch of even length [6]. An improvement of ranking in this case is defined as $a^{\mathcal{F}'} \succ a^{\mathcal{F}}$, ranking a against itself in the different scenarios.

Principle 14 (Strict Addition of Defense Branch ($\oplus DB$) [6]). A ranking-based semantics θ complies with *Strict Addition of Defense Branch*, if the addition of a defense branch to a improves its ranking.

Principle 15 (Addition of Defense Branch ($+DB$) [6]). A ranking-based semantics θ complies with *Addition of Defense Branch*, if for any attacked argument a , the addition of a defense branch to a improves its ranking.

Principle 16 (Increase of Defense Branch ($\uparrow$ DB) [6]). A ranking-based semantics θ complies with *Increase of Defense Branch*, if for any argument a with a defense branch, increasing the length of the defense branch degrades a 's ranking.

Proposition 17 (Addition and Increase of Defense Branch).

The AttDel semantics does not comply with addition of defense branch or increase of defense branch.

Proof. This is demonstrated by the counter-example in Figure 14.

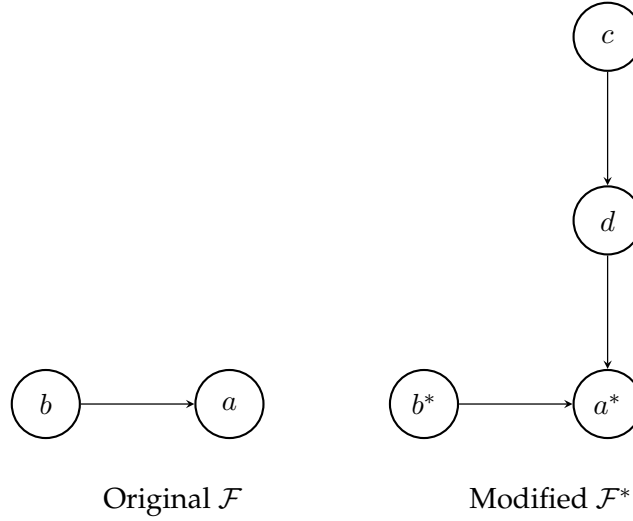


Figure 14: Comparison of an argument before and after the addition of a defense branch to disprove +DB.

The addition of a defense branch to any argument a in an AF $\mathcal{F}$ does not improve its ranking within the AttDel semantics. Disproving the non-strict version of this principle, that the addition of a defense branch to any attacked argument does not improve its ranking directly proves the non-compliance of the semantics for the strict version.

In the AFs $\mathcal{F}$ and $\mathcal{F}^*$ depicted in Figure 14, for the semantics to comply with addition of defense branch, the relationship $a^* \succ_{\mathcal{F}^*}^{AttDel} a$ would need to hold for a and its defense-branch added counter-image a^* . However, since deleting one attack makes a part of the grounded extension and a^* is not in the grounded extension, a^* also needs at least, in this case exactly, one attack deletion, to be included in the grounded extension. Therefore $a^* \succ_{\mathcal{F}^*}^{AttDel} a$ does not hold. Increasing the length of the defense branch would not change this result.

Thus, the AttDel semantics does not comply with addition of defense branch, strict addition of defense branch, or increase of defense branch. $\square$

Principle 17 (Addition of Attack Branch (+AB) [6]). A ranking-based semantics θ

complies with *Addition of Attack Branch*, if for any argument a , the addition of an attack branch to a degrades its ranking.

Proposition 18 (Addition of Attack Branch).

The AttDel semantics complies with addition of attack branch.

Proof. For our semantics, given an AF $\mathcal{F}$ and its modified version $\mathcal{F}'$ with an added attack branch of length $n > 0$ attacking a , this is satisfied if:

$$AttDelScore_{\mathcal{F}'}(a) > AttDelScore_{\mathcal{F}}(a)$$

Let $\mathcal{F} = (A, R)$ be an AF and $a \in A$ be an argument with $AttDelScore_{\mathcal{F}}(a) = x$. Let $\mathcal{F}' = (A', R')$ be the modified AF where $A' = A \cup AttBranch$, $A \cap AttBranch = \{a\}$, and $R' = R \cup \{(a_n, a_{n-1}), \dots, (a_1, a)\}$.

By definition, this attack branch attacks a and is not defended against, as it has no other connections to the rest of $\mathcal{F}$. Since a is undefended, it cannot be part of the grounded extension. To make a defended, at least one attack from the new attack branch must be deleted. Consequently, $AttDelScore_{\mathcal{F}}(a) = x$ implies

$$AttDelScore_{\mathcal{F}'}(a) = x + 1.$$

Thus, the AttDel semantics complies with addition of attack branch. $\square$

Principle 18 (Increase of Attack Branch ($\uparrow AB$) [6]). A ranking-based semantics θ complies with *Increase of Attack Branch*, if for any argument a with an attack branch, increasing the length of the attack branch improves a 's ranking.

Proposition 19 (Increase of Attack Branch).

The AttDel semantics does not comply with increase of attack branch.

Proof. Since the length of the attack branch has no influence on the increase of $AttDelScore$, as for any attack branch the increase in $AttDelScore$ is exactly 1, the ranking of a remains unchanged regardless of the branch length.

This proves that the AttDel semantics does not comply with increase of attack branch. $\square$

Principle 19 (Attack vs Full Defense (AvsFD) [6]). A ranking-based semantics complies with *Attack vs Full Defense* if an argument that is attacked by one non-attacked argument ranks lower than an argument that does not have an attack branch.

Proposition 20 (Attack vs Full Defense).

The AttDel semantics complies with attack vs full defense.

Proof. For our semantics, given an AF $\mathcal{F}$ where argument a has zero attack branches and argument b is attacked by exactly one non-attacked argument, this is satisfied if:

$$AttDelScore_{\mathcal{F}}(a) < AttDelScore_{\mathcal{F}}(b)$$

Let $\mathcal{F} = (A, R)$ be an AF containing two arguments, a and b . Let argument $a \in A$ possess some amount $n \in \mathbb{N}$ of defense branches and zero attack branches. Let

argument $b \in A$ be attacked by exactly one argument c , such that $R_1^-(b) = \{c\}$ and $R_1^-(c) = \emptyset$.

By the definition of the grounded extension, since a is either fully defended by non-attacked arguments or is itself unattacked, it is necessarily part of the grounded extension. Therefore:

$$AttDelScore_{\mathcal{F}}(a) = 0$$

Argument b is attacked by a single non-attacked argument c . Since c is unattacked, it is a member of the grounded extension. Because the grounded extension is conflict-free, b cannot be accepted by the grounded semantics in its current state. To make b part of the grounded extension, the attack from c must be deleted. Therefore:

$$AttDelScore_{\mathcal{F}}(b) > 0$$

Since a lower score implies a higher rank, it follows that $a \succ_{\mathcal{F}}^{AttDel} b$. Thus, the AttDel semantics complies with attack vs full defense. $\square$

4.14 Comparison Between Ranking-Based Semantics

Following the principle-based analysis, we now evaluate the Attack Deletion ranking-based semantics (θ^{AttDel}) within the context of the established literature. We utilize the comparative framework provided by Bonzon et al. [6] to benchmark our results against a variety of existing ranking-based semantics, interpreting the implications in a ranking that each principles compliance or non-compliance can give insight to. This comparison serves to highlight the unique behavioral profile of θ^{AttDel} .

The following ranking-based semantics are used for comparison, as formalized in the comparative study by Bonzon et al. [6]:

Ranking-based Semantics 1 (Social Abstract Argumentation (SAF) [16]). Rather than a single semantics, this represents a family of semantics where a model is a solution to an equation system, which can be implemented and weighed differently. Broadly, the semantics calculates a strength score for arguments by combining some initial score with an aggregation of scores of its direct attackers.

Ranking-based Semantics 2 (Categoriser (Cat) [4]). A semantics that recursively computes a value for each argument based on the idea that an attacker's strength is inversely proportional to the strength of its own attackers.

Ranking-based Semantics 3 (Burden-Based (Bbs) [1]). Similar to the categoriser, this semantics uses an iterative process to assign *burden numbers* to arguments, where an argument is considered weaker if its attackers are themselves relatively unburdened.

Ranking-based Semantics 4 (Discussion-Based (Dbs) [1]). This semantics ranks arguments by counting paths ending in the argument to score, punishing arguments with more path ends. On equal results between two arguments, the semantics recursively increases the path length, alternating between rewarding defense paths and punishing attack paths, until a difference is found.

Ranking-based Semantics 5 (Tuples* [7]). This semantics evaluates an argument by lexicographically comparing the lengths of all its ancestor branches. It prioritizes arguments with shorter defense branches and longer attack branches.

Ranking-based Semantics 6 (Matt & Toni (M&T) [17]). Based on game theory, this semantics determines strength by calculating the value of a zero-sum game between a proponent and an opponent of a given argument to score, where the payoff is determined by the number of successful attacks between their chosen sets of arguments.

Ranking-based Semantics 7 (Grounded (Gr) [6, 9]). This serves as a baseline ranking that provides a binary classification following the grounded semantics, ranking all arguments in the grounded extension as equally acceptable and all others as equally rejected.

While *gr*-compatibility has not yet been established for many ranking-based semantics, it has been proven that Cat, Dbs, Bbs, and M&T do not comply with this principle [20]. By the definition of σ -compatibility, Tuples* cannot be *gr*-compatible. The principle requires the comparability of any two arguments [5], as defined in total, which Tuples* does not comply with [6]. The full comparative results of the principle analysis between different semantics are shown in Table 1.

Properties	SAF	Cat	Dbs	Bbs	Tuples*	M&T	Grounded	AttDel
Abs	✓	✓	✓	✓	✓	✓	✓	✓
In	✓	✓	✓	✓	✓	✓	✓	✓
Tot	✓	✓	✓	✓	×	✓	✓	✓
NaE	✓	✓	✓	✓	✓	✓	✓	✓
VP	✓	✓	✓	✓	✓	✓	×	×
DP	✓	✓	✓	✓	×	×	×	×
CT	✓	✓	✓	✓	×	×	✓	×
SCT	✓	✓	✓	✓	×	×	×	×
CP	×	×	✓	✓	×	×	×	×
QP	×	×	×	×	×	×	✓	×
DDP	×	×	×	✓	×	×	×	×
SC	×	×	×	×	-	✓	×	×
$\oplus$ DB	×	×	×	×	×	×	×	×
+DB	×	×	×	×	✓	×	×	×
$\uparrow$ AB	✓	✓	✓	✓	✓	×	×	×
$\uparrow$ DB	✓	✓	✓	✓	✓	×	×	×
+AB	✓	✓	✓	✓	✓	✓	×	✓
AvsFD	×	×	×	×	✓	✓	✓	✓
<i>gr</i> -Comp	?	×	×	×	×	×	✓	✓

Table 1: Properties Comparison including Attack Deletion Semantics

4.15 Interpretation and Discussion of Results

Bonzon et al. [6] have classified the introduced principles into two different categories. The first category, consisting of Abs, In, VP, NaE, Tot and +AB, are semantics that serve as a baseline of what should be satisfied by all semantics. The other principles aim to distinguish different properties between ranking-based semantics in detail. The inclusion of the grounded extension is meant to make an example of the principle profile of bad ranking-based semantics, which is the case for the grounded extension, since lacks graduality in its ranking, only classifying between accepting and rejecting. It is notable that this significant drawback does not go past AttDel, since its granular ranking between different arguments only applies to arguments that are not part of the grounded extension. This is the reason why VP is not satisfied by AttDel, and is an issue that must be addressed, as arguments that are fully unattacked intuitively should rank higher than arguments that are attacked [6]. The current AttDel treats arguments accepted by the grounded extension as perfect, which does have some merit, because under the grounded extension, accepted arguments are deemed as uncontroversially acceptable [9]. Going by the logic of AttDel itself, attack deletions are a valid tool for establishing rankings between arguments. Following up this logic, an attacked argument, that is defended by the grounded extension can be made unaccepted by deleting defensive attacks. For an argument that is unattacked, this is not possible. As this logic can be applied, non-compliance with VP is not a necessary drawback for the semantics, but more of a symptom of inconsistent logic that can be improved upon.

AttDel can be classified into the same class of semantics as Tuples* and M&T. These semantics do not factor the local surroundings of an argument differently to further away parts of the AF. These semantics satisfy attack vs full defense, as they globally take whole branches of into account, not prematurely punishing defended arguments. Conversely, other semantics primarily score based on the direct surroundings of the argument, giving decreasing value based on the distance from the argument.

This fundamental divergence in scoring mechanics the two classes of semantics hard to compare. While local semantics rely on damping factors to diminish the impact of distant attackers, AttDel views the framework as a global optimization problem where an attacker's status is prioritized over its distance.

Consequently, the following analysis focuses on its closest relatives in the class complying with attack vs full defense: Tuples* and M&T. Comparing these allows for a better characterization of how AttDel behaves and can show potential benefits compared to similar semantics.

Properties	Tuples*	M&T	AttDel
Abs	✓	✓	✓
In	✓	✓	✓
Tot	×	✓	✓
NaE	✓	✓	✓
VP	✓	✓	×
SC	-	✓	×
+DB	✓	×	×
↑AB	✓	×	×
↑DB	✓	×	×
+AB	✓	✓	✓
AvsFD	✓	✓	✓
<i>gr</i> -Comp	×	×	✓

Table 2: Global Ranking-based Semantics Principle Differences

As seen in Table 2 we find, out of the compared similar semantics, AttDel is the only one to comply with *gr*-compatibility, which is a clear advantage compared to the other semantics, as it bridges the gap between the simple extension-based semantics and a more detailed ranking-based semantics. This compliance captures the purpose of the semantics and is the reason for its intuitiveness. On a purely quantitative scale, however, AttDel complies with fewer principles than M&T and Tuples, as it does not comply with self contradiction or many of the change based principles. Going past purely principle based analysis, compared to M&T, AttDel excels in intuitiveness, as the semantics can be easily explained and results, including sets of deleted attacks, are easily tracable. M&T, dealing with probability distributions and mixed strategies [17] is more complex. Compared to Tuples*, AttDel’s biggest advantage is its compliance with Total. A semantics that does not comply with Total makes it hard to use in practical applications [6]. An altered variant of AttDel that complies with void precedence, while preserving the compliance with the other principles, would make the semantics viable under all necessary principles [6] and would solidify its advantage regarding *gr*-compatibility within an acceptable ranking-based semantics.

5 Extended Attack Deletion Ranking-Based Semantics

The principle analysis of AttDel has shown the promising result of uniquely complying with the grounded extension, which the other semantics fail to do. There were also the finding of a significant drawback, as AttDel does not comply with void precedence. In this section, we propose an extended AttDel ranking-based semantics that complies with VP while preserving the unique profile of the semantics.

The reason for the non compliance with VP is the fact that all arguments that are in the grounded extension of an AF receive an *AttDelScore* of 0. This includes attacked arguments and unattacked arguments. An additional ranking of all arguments within the grounded extension that ranks unattacked arguments higher than attacked arguments, while preserving the original semantics ranking over arguments outside of the grounded extension would cause this new semantics to comply with VP.

AttDel measures the distance to the grounded extension of arguments, ranking arguments that are very close to the grounded extension, requiring only few attack removals to be part of it, higher than ones far away. Continuing the approach to within the grounded extension, we can measure the minimum number of attacks to be deleted for an accepted argument to leave the grounded extension. This is a novel alteration of extension enforcement, where the goal, opposite to traditional extension enforcement, is to modify the AF in a way that an argument is denied by the semantics. This new approach using modifications on accepted arguments allows for the evaluation of the strength of an arguments defense. We introduce a new scoring function that ranks attacked arguments within the grounded extension:

Definition 14 (Attack Deletion Score for Denial by σ^{gr} (*AttDelDenyScore*)). Given an AF $\mathcal{F} = (A, R)$ with an argument $a \in A$, where there exists $b \in A$ and $(b, a) \in R$ and a predicate

$$Gr(\mathcal{F}', a) = \{a \text{ is in the grounded extension of } \mathcal{F}'\}$$

find a minimal k for which a subset $R' \subseteq R$ exists such that $|R'| = k$, $\mathcal{F}' = (A, R \setminus R')$ and $Gr(\mathcal{F}', a)$ is *false*.

Given this score, arguments within the grounded extension can be ranked based on their resilience within the grounded extension. If an argument a is in the grounded extension and is attacked by argument b , given the formal definition of grounded semantics, there exists an argument c within the grounded extension that attacks b , defending a [9]. Deleting the defending attack, given there are no other defenders defending a from b , causes a to leave the grounded extension. Provided multiple defenders, the minimum number of attack deletion causing a to be undefended from at least one attacker, is the *AttDelDenyScore* of the argument, with a higher score implying a more resilient, stronger argument. Given the new score, we introduce the new unified *ModAttDelScore*.

Definition 15 (Modified Attack Deletion Score for σ^{gr} *ModAttDelScore*). Given an AF $\mathcal{F} = (A, R)$ with an argument $a \in A$ and a predicate

$$UnAtt(\mathcal{F}, a) = \{\{a\}^- = \emptyset\}$$

we introduce a new score $ModAttDelScore : A \rightarrow (\mathbb{Z} \setminus \{0\}) \cup \{-\infty\}$:

$$ModAttDelScore_{\mathcal{F}}(a) = \begin{cases} -\infty & \text{if } UnAtt(\mathcal{F}, a) \\ AttDelScore_{\mathcal{F}}(a) - AttDelDenyScore_{\mathcal{F}}(a) & \text{otherwise} \end{cases}$$

The usage of $-\infty$ ensures the higher ranking of unattacked arguments for complying with VP, symbolizing the highest possible rank. Subtracting $AttDelDenyScore$ from $AttDelScore$ yields a strict classification between arguments based on their grounded acceptance. Because an argument cannot simultaneously require effort to be enforced and effort to be denied, the two underlying scores are mutually exclusive regarding a value of 0. With one of the scores always being 0 and the other being positive, $ModAttDelScore$ cannot be 0. This non-zero property ensures the strict differentiation of the grounded acceptance based on the sign of the score, with $ModAttDelScores$ of arguments inside of the grounded extension being negative and $ModAttDelScores$ of arguments outside of the grounded extension being positive.

Lemma 1 (Score Mirroring).

Let $\mathcal{F} = (A, R)$ be an argumentation framework and $a \in A$.

1. If $a \notin \sigma_{\mathcal{F}}^{gr}$, then $ModAttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(a)$.
2. If $a \in \sigma_{\mathcal{F}}^{gr}$, then $ModAttDelScore_{\mathcal{F}}(a) < 0$.

Proof. By definition of $AttDel$, $a \in \sigma_{\mathcal{F}}^{gr} \iff AttDelScore_{\mathcal{F}}(a) = 0$.

If $a \notin \sigma_{\mathcal{F}}^{gr}$, a is not unattacked and zero attack deletions are required to deny it from the grounded extension, meaning $AttDelDenyScore_{\mathcal{F}}(a) = 0$. Applying the definition yields:

$$ModAttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(a) - 0 = AttDelScore_{\mathcal{F}}(a)$$

If $a \in \sigma_{\mathcal{F}}^{gr}$, it requires zero enforcement deletions, meaning $AttDelScore_{\mathcal{F}}(a) = 0$. If a is unattacked, its modified score is defined as $-\infty$ and $-\infty < 0$. If a is attacked, successfully denying it requires deleting at least one defending attack, meaning $AttDelDenyScore_{\mathcal{F}}(a) \geq 1$. Applying the definition yields:

$$ModAttDelScore_{\mathcal{F}}(a) = 0 - AttDelDenyScore_{\mathcal{F}}(a) \leq -1 < 0$$

□

Definition 16 (Modified Attack Deletion Ranking-Based Semantics ($ModAttDel$)).

Given an AF $\mathcal{F} = (A, R)$, define a ranking-based semantics $\theta^{ModAttDel}$. For all $a, b \in A$:

$$a \succeq_{\mathcal{F}}^{\theta^{ModAttDel}} b \iff ModAttDelScore_{\mathcal{F}}(a) \leq ModAttDelScore_{\mathcal{F}}(b)$$

Example 4 (Comparison of AttDel and ModAttDel Rankings). Consider the argumentation framework $\mathcal{F}$ depicted in Figure 15. It consists of four target arguments to be evaluated: u (which is unattacked), a (strongly defended by y_1 and y_2), b (defended by a single chain), and c (attacked by an unattacked argument, thus not in the grounded extension).

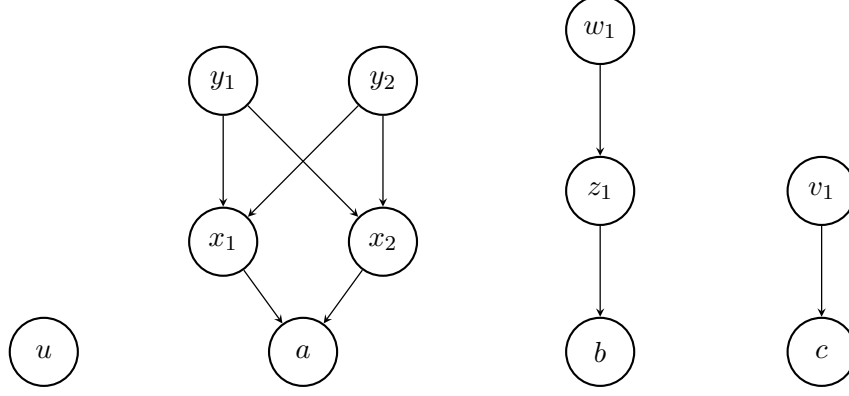


Figure 15: Example AF $\mathcal{F}$ comparing structural resilience.

Under the original *AttDel* semantics, the distance to the grounded extension is measured solely by attack deletions for extension enforcement. Because u , a , and b are all inherently part of the grounded extension, they all require zero attack deletions to be enforced. Argument c requires one deletion (the attack (v_1, c)) to become grounded. The resulting scores are:

- $AttDelScore(u) = 0$
- $AttDelScore(a) = 0$
- $AttDelScore(b) = 0$
- $AttDelScore(c) = 1$

This results in the ranking:

$$u \simeq_{\mathcal{F}}^{\theta^{AttDel}} a \simeq_{\mathcal{F}}^{\theta^{AttDel}} b \succ_{\mathcal{F}}^{\theta^{AttDel}} c$$

This demonstrates the failure to satisfy VP, as the unattacked argument u is ranked equally with the attacked arguments a and b . Furthermore, the semantics fails to capture that a is structurally more robust than b .

By applying the new *ModAttDelScore*, we evaluate both the inclusion and denial distances. The modified scores are:

- $ModAttDelScore(u) = -\infty$
- $ModAttDelScore(a) = -2$

- $ModAttDelScore(b) = -1$
- $ModAttDelScore(c) = 1$

Yielding the strict ranking:

$$u \succ_{\mathcal{F}}^{\theta^{ModAttDel}} a \succ_{\mathcal{F}}^{\theta^{ModAttDel}} b \succ_{\mathcal{F}}^{\theta^{ModAttDel}} c$$

This example illustrates how *ModAttDel* successfully enforces Void Precedence by isolating unattacked arguments at the top of the ranking. It also shows finer granularity in the ranking, allowing for a more accurate assessment of argument strength.

5.1 Principle Analysis

The *ModAttDel* semantics features a more granular ranking mechanism, while keeping the same general order of arguments as *AttDel*. In this section, we analyze which principles are kept from *AttDel* and in which principles the two semantics differ. The compliance of many principles by *AttDel* has been disproven using examples comparing *AttDelScores* of arguments outside of the grounded extension. Since $ModAttDelScore = AttDelScore$ for all arguments rejected by the grounded semantics, as shown in Lemma 1, these counter-examples still hold for the new semantics. Consequently, *ModAttDel* does not comply with QP, SC, DP, DDP, DB, +DB, $\oplus$ DB, $\uparrow$ DB, $\uparrow$ AB, *adm*-compatibility, *co*-compatibility, *pr*-compatibility and *st*-compatibility. We start this analysis with Void Precedence, as this is the principle that motivated introducing a new semantics.

5.1.1 Void Precedence

Proposition 21 (Void Precedence).

ModAttDel complies with VP, ranking all unattacked arguments higher than all attacked arguments.

Proof. Let $AF \mathcal{F} = (A, R)$ and $a, b \in A$, $\{a\}_{\mathcal{F}}^- = \emptyset$ and $\{b\}_{\mathcal{F}}^- \neq \emptyset$. We find:

$$ModAttDelScore_{\mathcal{F}}(a) = -\infty$$

and

$$ModAttDelScore_{\mathcal{F}}(b) \geq -(|R| - 1)$$

because causing denial of b by the grounded semantics at most requires deletion of all attacks except one direct attacker of b . In the case where b is not part of the grounded extension, $ModAttDelScore_{\mathcal{F}}(b) > 0$, therefore $-(|R| - 1)$ is the smallest possible *ModAttDelScore* of b . Since $|R|$ is, by definition, finite [9], $ModAttDelScore_{\mathcal{F}}(b) > -\infty$ and $a \succ_{\mathcal{F}}^{ModAttDel} b$. $\square$

We continue the analysis of the remaining principles.

5.1.2 Abstraction

Proposition 22 (Abstraction).

The ModAttDel semantics complies with abstraction.

Proof. This holds if for any isomorphic argumentation frameworks $\mathcal{F} = (A, R)$ and $\mathcal{F}' = (A', R')$ mapped by an isomorphism γ , the following holds for any $a \in A$:

$$ModAttDelScore_{\mathcal{F}}(a) = ModAttDelScore_{\mathcal{F}'}(\gamma(a))$$

Having already proven the compliance of abstraction of $AttDelScore$ previously, it remains to be proven, for any attacked argument $a' \in A$:

$$AttDelDenyScore_{\mathcal{F}}(a') = AttDelDenyScore_{\mathcal{F}'}(\gamma(a'))$$

and for any unattacked argument $a'' \in A$:

$$ModAttDelScore_{\mathcal{F}}(a'') = ModAttDelScore_{\mathcal{F}'}(\gamma(a'')) = -\infty$$

Let $\mathcal{F} = (A, R)$ and $\mathcal{F}' = (A', R') = (\gamma(A), \gamma(R))$ be isomorphic under γ . We prove that for any $x \in A$ and its image $x' = \gamma(x) \in A'$, where $\{x\}^- \neq \emptyset$, it holds that $AttDelDenyScore_{\mathcal{F}}(x) = AttDelDenyScore_{\mathcal{F}'}(x')$.

Given $AttDelDenyScore_{\mathcal{F}}(x) = k$, there exists a set $R^{mod} \subseteq R$ where $|R^{mod}| = k$ such that x is not in the grounded extension of $\mathcal{F}^{mod} = (A, R \setminus R^{mod})$. Since the grounded extension satisfies the abstraction principle as it is solely defined by the graph structure [9], if $\sigma_{\mathcal{F}}^{gr}$ is the grounded extension of $\mathcal{F}$, then $\gamma(\sigma_{\mathcal{F}}^{gr})$ is the grounded extension of $\mathcal{F}'$.

The isomorphic graph structure implies that x' is not in the grounded extension of $\mathcal{F}'^{mod} = (A', R' \setminus \gamma(R^{mod}))$. Since isomorphism is a bijective relation, $|R^{mod}| = |\gamma(R^{mod})|$. Furthermore, since γ is reversible, the removal of no smaller set than $|\gamma(R^{mod})|$ can enforce rejection of x' by the grounded semantics of $\mathcal{F}'$, as its inverse would cause a smaller set than R^{mod} to enforce the grounded extension for x , contradicting minimality.

Consequently, $AttDelDenyScore_{\mathcal{F}}(x) = AttDelDenyScore_{\mathcal{F}'}(\gamma(x))$.

We find, for any attacked argument $a' \in A$,

$$ModAttDelScore_{\mathcal{F}}(a) = ModAttDelScore_{\mathcal{F}'}(\gamma(a))$$

because $ModAttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(a) - AttDelDenyScore_{\mathcal{F}}(a)$ and $AttDelScore$ and $AttDelDenyScore$ comply with Abstraction.

Let $u \in A$ be an unattacked argument. By definition, $ModAttDelScore_{\mathcal{F}}(u) = -\infty$. Given the attack relation structure is preserved by an isomorphism, $\gamma(u)$ is unattacked and $ModAttDelScore_{\mathcal{F}}(\gamma(u)) = -\infty$

Because the scores are identical across the isomorphism, for attacked arguments, as well as unattacked arguments, the ranking defined by the semantics is preserved. Thus, the ModAttDel semantics complies with abstraction. $\square$

5.1.3 Independence

Proposition 23 (Independence).

The ModAttDel semantics complies with independence.

Proof. We have already proven the compliance of AttDel by proving the equality of scores within connected components compared to the global AF. As ModAttDel mirrors AttDel for all arguments outside the grounded extension, this proof remains valid. Therefore, it only remains to prove the compliance of Independence for unattacked arguments and other arguments within the grounded extension.

Let $\mathcal{F} = (A, R)$ be an AF and $\mathcal{F}' = (A', R') \in CC(\mathcal{F})$ be a connected component containing an argument $a \in A'$.

If a is unattacked in $\mathcal{F}'$, by definition of a connected component, there are no attacks from outside A' into A' , meaning a is also unattacked in $\mathcal{F}$. Therefore, $ModAttDelScore_{\mathcal{F}}(a) = ModAttDelScore_{\mathcal{F}'}(a) = -\infty$.

If a is attacked and in the grounded extension, we must prove that the minimal set of attack deletions R'^{-} required to deny a from the grounded extension of $\mathcal{F}'$ is identical to the minimal set R^{-} required to deny a from the grounded extension of $\mathcal{F}$.

By the directionality principle of the grounded extension [3], the acceptability of an argument depends exclusively on the acceptability of its ancestors. By definition of a connected component, there exists no attack $r = (x, y) \in R$ such that $x \notin A'$ and $y \in A'$. Therefore, the grounded acceptance (and consequently, the denial) of a is completely isolated from any graph structure outside of $\mathcal{F}'$.

Consequently, any set of attack deletions R'^{-} that removes a from the grounded extension of the component $\mathcal{F}'^{mod} = (A', R' \setminus R'^{-})$ inherently removes a from the grounded extension of the full framework $\mathcal{F}^{mod} = (A, R \setminus R'^{-})$.

To prove minimality, assume there exists a set $R^{-} \subseteq R$ such that $|R^{-}| < |R'^{-}|$ and a is not in the grounded extension of $(A, R \setminus R^{-})$. Due to the directionality principle, any deletion in R^{-} involving arguments outside of A' is strictly redundant for the denial of a . This implies that a purely internal subset exists that is smaller than R'^{-} , which directly contradicts the premise that R'^{-} is minimal for $\mathcal{F}'$.

Consequently, $AttDelDenyScore_{\mathcal{F}}(a) = AttDelDenyScore_{\mathcal{F}'}(a)$. Because, for attacked arguments,

$$ModAttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(a) - AttDelDenyScore_{\mathcal{F}}(a)$$

and both underlying scores comply with independence, the score of any attacked argument is identical whether evaluated globally or within its isolated component, preserving the internal ranking globally. Additionally, any unattacked argument has the identical $ModAttDelScore = -\infty$.

Thus, the ModAttDel semantics complies with independence. $\square$

5.1.4 Total

Proposition 24 (Total).

ModAttDel complies with total.

Proof. As already shown that AttDel complies with total, implying an *AttDelScore* for all arguments. For full coverage by ModAttDelScore it remains to prove that all arguments can either be assigned an *AttDelDenyScore* if attacked or their *ModAttDelScore* is $-\infty$ if unattacked. Let $\mathcal{F} = (A, R)$ and $a \in A$ be an argument outside of the grounded extension. By definition, $\text{AttDelDenyScore}_{\mathcal{F}}(a) = 0$, as no argument deletion is required because a is not part of the grounded extension. Now let a be an attacked argument and part of the grounded extension. Let $b \in \{a\}^-$. The grounded extension defends a against b by virtue of the characteristic function, implying there being a non empty set of arguments within the grounded extension $D \subseteq \sigma_{\mathcal{F}}^{gr}$ and $D \subseteq \{b\}^-$, inherently implying that b is attacked. Removing all incoming attacks of b causes b to be part of the grounded extension. Since the grounded extension is conflict free, a is consequently rejected by the grounded extension, as b attacks a . Therefore all attacked arguments can receive an *AttDelDenyScore*. Since all attacked arguments can be assigned an *AttDelScore* and *AttDelDenyScore*, they can also be assigned a *ModAttDelScore*, which is the difference of the two scores. Unattacked arguments $u \in A$, by definition are assigned $\text{ModAttDelScore}_{\mathcal{F}}(u) = -\infty$. Therefore, all arguments can be assigned a *ModAttDelScore* mapping to $\{-\infty\} \cup \mathbb{Z}$. Since this set is totally ordered, any two arguments can be compared, proving ModAttDel complying with total. $\square$

5.1.5 Non-Attacked Equivalence

Proposition 25 (Non-Attacked Equivalence).

ModAttDel complies with NaE. All non-attacked arguments are ranked the same.

Proof. Let $\mathcal{F} = (A, R)$ be an AF with any two arguments $a, b \in A$, which are non-attacked. By definition, $\text{ModAttDelScore}_{\mathcal{F}}(a) = -\infty$ and $\text{ModAttDelScore}_{\mathcal{F}}(b) = -\infty$. Therefore $a \simeq_{\mathcal{F}}^{\text{ModAttDel}} b$. Since a and b were chosen arbitrarily, this equivalence applies to all non-attacked arguments, proving compliance with the principle. $\square$

5.1.6 Counter-Transitivity

Proposition 26 (Counter-Transitivity).

ModAttDel does not comply with CT or SCT.

Proof. The proof of non-compliance of CT by AttDel relies on the direct attackers of arguments a and b in the AF $\mathcal{F}$ in Figure 9 having the same *AttDelScore* $_{\mathcal{F}}$. For all arguments $x \in \{a_1, a_2, b_1, b_2\}$, $\text{ModAttDelScore}_{\mathcal{F}}(x) = -1$, therefore, the direct attackers of a and b remain equally strong under ModAttDel and the same proofs follow for CT and SCT. $\square$

5.1.7 Cardinality Precedence

Proposition 27 (Cardinality Precedence).

The ModAttDel semantics does not comply with cardinality precedence.

Proof. This is demonstrated by the counter-example in Figure 16.

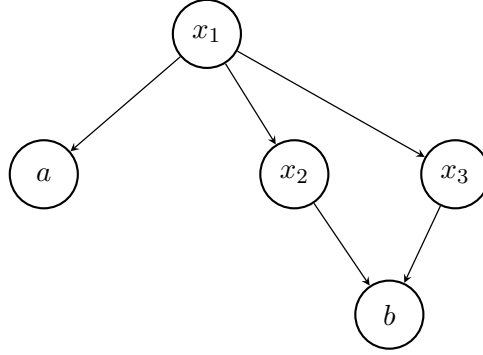


Figure 16: Counter-example AF $\mathcal{F}$ for Cardinality Precedence.

In the AF $\mathcal{F}$ depicted in Figure 16, a would be part of the grounded extension if its single incoming attack was deleted. b is already part of the grounded extension. To cause rejection by the grounded semantics for b , one of the attacks from x_1 towards its direct attackers require deletion. the relevant arguments therefore evaluate to the following modified attack deletion scores:

- $ModAttDelScore_{\mathcal{F}}(a) = 1$
- $ModAttDelScore_{\mathcal{F}}(b) = -1$

Based on these scores, $ModAttDelScore_{\mathcal{F}}(a) > ModAttDelScore_{\mathcal{F}}(b)$, which yields $b \succ_{\mathcal{F}}^{ModAttDel} a$. However, under cardinality precedence, because b has two direct attackers, whereas a only has one, $a \succ_{\mathcal{F}}^{ModAttDel} b$ is required.

This contradiction proves that the ModAttDel semantics does not comply with cardinality precedence. $\square$

5.1.8 grounded-Compatibility

Proposition 28 (gr-Compatibility).

ModAttDel complies with grounded compatibility.

Proof. As Lemma 1 shows, for any argument $a \in A$ in an AF $\mathcal{F} = (A, R)$ that is not part of the grounded extension of $\mathcal{F}$, $ModAttDelScore_{\mathcal{F}}(a) > 0$, and for any argument $b \in A$ that is part of the grounded extension of $\mathcal{F}$, $ModAttDelScore_{\mathcal{F}}(b) < 0$. Therefore, $ModAttDelScore_{\mathcal{F}}(a) > ModAttDelScore_{\mathcal{F}}(b)$, implying $b \succ_{\mathcal{F}}^{ModAttDel} a$. $\square$

5.1.9 Addition of Attack Branch

Proposition 29 (Addition of Attack Branch).

ModAttDel complies with addition of attack branch.

Proof. For an AF $\mathcal{F} = (A, R)$ with an argument a , and an AF $\mathcal{F}'$ which mirrors $\mathcal{F}$, containing an argument a' which mirrors a , but has an attack branch added. a' is not part of the grounded extension, because it has an undefended attack branch. If a is part of the grounded extension, due to ModAttDel complying with *gr*-Compatibility, a ranks higher than a' . If a is not part of the grounded extension, $ModAttDelScore_{\mathcal{F}}(a) = AttDelScore_{\mathcal{F}}(a)$. Because AttDel complies with addition of attack branch, ModAttDel complies with the principle for arguments outside the grounded extension. With the compliance of arguments accepted and rejected by the grounded extension, ModAttDel complies with addition of attack branch. $\square$

5.1.10 Attack vs Full Defense

Proposition 30 (Attack vs Full Defense).

ModAttDel complies with attack vs full defense.

Proof. By definition, an argument with zero attack branches is inherently accepted by the grounded extension, while an argument attacked by an argument without defense is rejected [9]. Because ModAttDel complies with *gr*-compatibility, it ranks any grounded argument higher than any non-grounded argument. ModAttDel therefore complies with attack vs full defense. $\square$

5.2 Improvement over AttDel

Having established ModAttDel and analyzed its compliance with the common ranking-based semantics principles, we can verify whether resolving the problem of non-compliance with VP has any effect on the compliance with other principles.

Properties	Tuples*	M&T	AttDel	ModAttDel
Abs	✓	✓	✓	✓
In	✓	✓	✓	✓
Tot	×	✓	✓	✓
NaE	✓	✓	✓	✓
VP	✓	✓	×	✓
SC	-	✓	×	×
+DB	✓	×	×	×
↑AB	✓	×	×	×
↑DB	✓	×	×	×
+AB	✓	✓	✓	✓
AvsFD	✓	✓	✓	✓
<i>gr</i> -Comp	×	×	✓	✓

Table 3: Global Ranking-based Semantics Principle Differences including ModAttDel

As shown in Table 3, ModAttDel mirrors the principle compliance of AttDel with the exception of VP. It preserves its unique compliance with the grounded extension. ModAttDel extends the methodology of AttDel to accepted arguments, ranking strongly defended arguments higher than weakly defended ones. Since AttDel postulates attack deletion as a valid evaluative tool, the expansion by ModAttDel is a logically sound measure. The compliance of VP completes the compliance of all important principles [6] making ModAttDel a viable semantics.

6 Implementation

As a proof of concept of plausibility that the ranking-based semantics can be computed efficiently, we introduce an approach of implementation. Following this, we propose optimizations that may mitigate weaknesses that arise in specific use cases of AttDel. The calculation is computationally expensive, as determining the minimal attack deletions necessitates evaluating the grounded extension of the full argumentation framework for each possible subset of attacks. As this problem is a version of *non-strict extension enforcement under grounded semantics* with a single element input set, it is also NP-hard [18]. A first imperative solution illustrates the need for an optimized approach of calculation.

6.1 Primitive Solution

Given argument ax within the connected component $\mathcal{F}' = (A', R')$ in AF $\mathcal{F} = (A, R)$ with a function $\text{ground}((A, R'))$ that calculates the grounded extension for an AF $\mathcal{F} = (A, R')$ in polynomial time [10] and a function $\text{subset_of_size}(\text{set},$

size) returning all subsets of a given set at a given size, the score can be computed as follows:

```

deletion_score(A', R') :
  for i=0 to i=|R'|:
    for attacks_to_remove in subsets_of_size(R', i):
      attacks_to_check = R' \ attacks_to_remove
      if (ground(A, attacks_to_check).contains(AX)) :
        return |attacks_to_remove|

```

By effectively iterating through the power set sorted from smallest to largest, the optimal solution is found at the first occurrence of AX in the given grounded extension. Since for a set of size n , the powerset is of size 2^n , this algorithm is of exponential complexity, which illustrates the need for optimization.

6.2 ASP Solution

We introduce a function for scoring arguments based on the minimum attack deletions to enforce a grounded extension. Niskanen et al. [18] have laid the groundwork for this with different implementations of extension enforcement.

Definition 17 (Extension Enforcement). Given an extension semantics σ and an AF $\mathcal{F} = (A, R)$ and a set $A' \subseteq A$, find the minimum set $R' \subseteq R$ such that A' is the σ -extension of $\mathcal{F}' = (A, R \setminus R')$. With the *non-strict extension enforcement*, A' must be a subset of the σ -extension of $\mathcal{F}'$ [18].

The implementation for our scoring function calculates non-strict extension enforcement with single element input sets and returns the size of the resulting set. The preexisting implementations can therefore be used verbatim. Niskanen et al. [18] define a language-agnostic ASP approach of implementation. Our implementation leverages the language features of Clingo to provide a shortened version.²

6.2.1 ASP Implementation using the Grounded Extension

For a given set of arguments and attacks, represented as the facts $\text{arg}(X)$ and $\text{att}(X, Y)$ and an enforced argument a , this ASP implementation solves for the smallest set of attack deletions to make a part of the grounded extension. The logic of the implementation consists of three major parts, the definition of the AF and the targeted argument, the calculation of the membership of the grounded extension and the minimization of the set of attacks to delete.

To calculate the score, the whole AF must be encoded as facts in the ASP program.

²The complete source code and benchmark scripts used in this section are publicly available at <https://github.com/JakobHaastert/master-thesis-resources>.

```
arg(a) .                                     (1)
arg(b) .
arg(c) .
```

```
att(a,b) .                                  (2)
att(b,c) .
```

```
enf(a) .                                    (3)
```

This example set of argument (1) and attack (2) facts encodes the AF $\mathcal{F}$ from Figure 17

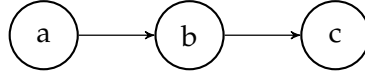


Figure 17: Argumentation framework encoded by Clingo program.

`enf(a)` . defines the argument to be enforced in the grounded extension.

The following module of the program is responsible for minimizing the amount of attack deletions to make the enforced argument part of the grounded extension.

```
{ att_to_remove(X,Y) : att(X,Y) }.          (4)
```

```
mAtt(X,Y) :- att(X,Y), not att_to_remove(X,Y). (5)
```

```
:- not grounded(X), enf(X).                  (6)
```

```
#minimize { 1,X,Y : att_to_remove(X,Y) }.    (7)
```

The program chooses a subset of attacks to remove (4). The program optimizes for minimality of this set (7). From the removed attacks, a modified attack relation is generated symbolizing the attacks that remain after removal (5). The grounded extension is then constrained to contain the enforced argument (6).

The section of the program calculating the grounded extension utilizes Clingos ability to utilize conditional literals and is based on the ASPARTIX system [11].

```
grounded(X) :- arg(X), defeated(Y) : mAtt(Y,X). (8)
```

```
defeated(X) :- arg(X), mAtt(Y,X), grounded(Y). (9)
```

The implementation calculates the grounded extension for the resulting AF after deleting attacks, and therefore uses the `mAtt` relation as its attack relation. The program defines all arguments to be grounded if all their attackers are defeated (8). Attackers are deemed as defeated if they are attacked by the grounded extension (9). This is an encoding of the definition of the grounded extension, which is defined

by iterating over defended arguments using the characteristic function until the least fixed point is found:

$$\Delta_{\mathcal{F}}(S) = \{a \in A \mid a \text{ is defended by } S\}$$

$$GE = \bigcup_{i \geq 0} \Delta_{\mathcal{F}}^i(\emptyset)$$

Using `#show att_to_remove/2.`, the Clingo program returns a minimal set of attacks to remove, which, by calculating its cardinality using a script yields the desired $AttDelScore_{\mathcal{F}}(a)$.

6.2.2 ASP Implementation using Strongly Admissible Sets

The previous implementation determines the enforced arguments belonging in the grounded extension by calculating the whole grounded extension. The fully calculated grounded extension is not further used for the purpose of score calculation, which leads to the assumption that some performance is wasted calculating other parts of the grounded extension that are unrelated to the targeted argument. Strongly admissible sets are a common approach of determining whether an argument is part of the grounded extension without calculating the whole extension [12].

Definition 18 (Strongly Admissible Set [12]). In an AF $\mathcal{F} = (A, R)$, a set of arguments $S \subseteq A$ is strongly admissible if S strongly defends all of its arguments.

An argument $a \in A$ is strongly defended by S if all of its attackers $\{a\}^-$ are attacked by an argument $b \in S \setminus \{a\}$ and b is strongly defended by $S \setminus \{a\}$.

In contrast to admissible sets, which are conflict-free sets, where each incoming attack is defended by the set, strongly admissible sets require the defense branches to be rooted in an unattacked argument [12].

If an argument is part of a strongly admissible set, it is also part of the grounded extension [12].

An implementation of AttDel using strongly admissible sets builds upon the same structure as the implementation using the grounded extension.

```
{ att_to_remove(X,Y) : att(X,Y) }.
```

```
mAtt(X,Y) :- att(X,Y), not att_to_remove(X,Y).
```



```
:- not strongly_admissible(X), enf(X).
```

(10)


```
#minimize { 1,X,Y : att_to_remove(X,Y) }.
```

```
#show att_to_remove/2.
```

The minimization implementation specifies that the enforced argument must be in a strongly admissible set (10).

The calculation of the strongly admissible sets is also based on the ASPARTIX system [11].

```

in(X) :- arg(X), not out(X).
out(X) :- arg(X), not in(X).

```

(11)

```

fixedPoint(X) :- in(X), defeated(Y) : mAtt(Y, X).
defeated(X) :- arg(X), fixedPoint(Y), mAtt(Y, X).

```

(12)

```

:- in(X), not fixedPoint(X).

```

```

strongly_admissible(X) :- in(X).

```

The solution guesses a set of `in` arguments, which will form the strongly admissible sets (11). The fixed point for each set is then calculated by using the same fixed point approach as the grounded extension (12), where defeat of attacking arguments of `in` is checked. The two implementation approaches perform very similar calculations with the difference, that the grounded extension approach uses a mechanism to check if the enforced argument is inside the grounded extension using a deterministic calculation of the grounded extension. This may be inefficient when the grounded extension is very large. The strongly admissible set approach non deterministically checks the membership of the enforced argument by guessing strongly admissible sets. If the enforced argument is inside one of these strongly admissible sets, the calculation may be much more efficient than the calculation of the grounded extension as a whole, because a strongly-admissible set is only a subset of the grounded extension [12].

In cases, where $AttDelScore > 0$, the solution candidates, where the enforced argument is not part of the grounded extension, typically makes up the computational load in non-trivial examples. Optimizing for efficiency in cases when a potential solution is found should therefore be of lower priority than the dismissal of non-suitable candidates. We can see that a singular attempt of finding a strongly admissible set containing the enforced argument can theoretically be done much faster than any such attempt using the grounded extension. This may seem intriguing. However the two approaches in the context of a whole optimization problem scale differently. The approach of computing the grounded extension to dismiss a solution candidate is much more suitable because it is a deterministic polynomial-complexity calculation [10]. Disproving the membership in the grounded extension this way can therefore be done efficiently. In contrast, the second approach must prove a negative: that no strongly admissible set containing the target exists. This requires an exhaustive search of the power set of relevant arguments, which is inherently exponential. This theoretical divergence in complexity suggests that the polynomial grounded approach performs significantly better in practice. However, because ASP solvers like Clingo are highly optimized for complex search and repetitive calculations [14] confirming or denying that this theoretical advantage translates into real improvements in performance necessitates a practical analysis.

6.2.3 Search Space Optimization

To assess whether the enforced argument is part of the grounded extension, it is not necessary to calculate the whole grounded extension. Arguments that are not connected to the enforced argument by an attack chain have no influence on its membership in the grounded extension, since they neither attack nor defend the argument itself or any of its defenders.

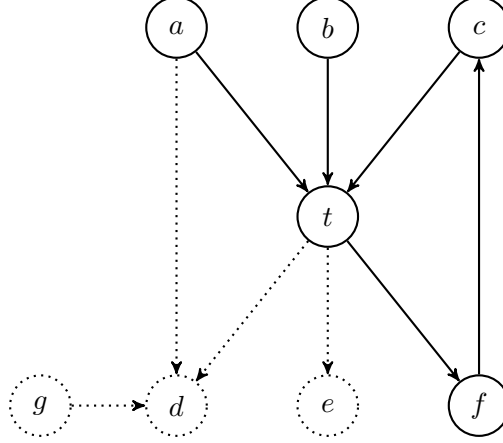


Figure 18: AF $\mathcal{F} = (A, R)$ pruned for efficient calculation of the membership of target argument t in the grounded extension.

The dotted attacks and arguments in Figure 18, e , d and g have no attacking branch towards t . Let $A' = A \setminus \{d, e, g\}$ be the set of relevant arguments. t is in the grounded extension of $\mathcal{F}$ iff it is in the grounded extension of $\mathcal{F}' = (A', R \cap (A' \times A'))$. As an optimization of our implementation of *AttDelScore*, we can therefore perform calculations on an alternative AF containing only relevant arguments and attacks. For this, we introduce the $\text{rArg}(X)$ predicate, which is a subset of $\text{arg}(X)$ containing only relevant arguments. $\text{rAtt}(X, Y)$ is a subset of $\text{att}(X, Y)$ of all relevant attacks.

$$\text{rArg}(X) \quad :- \quad \text{enf}(X) . \quad (13)$$

$$\text{rArg}(Y) \quad :- \quad \text{rArg}(X) , \text{att}(Y, X) . \quad (14)$$

$$\text{rAtt}(X, Y) \quad :- \quad \text{att}(X, Y) , \text{rArg}(X) , \text{rArg}(Y) . \quad (15)$$

The implementation specifies the enforced argument as the first relevant argument (13) and then builds a cone of relevant arguments based on the attack relation towards relevant arguments (14). The set of relevant attacks is then built by checking whether both arguments in an attack relation are relevant (15).

By substituting att for rAtt and arg for rArg in the calculation of the grounded extension and strongly admissible sets respectively, the search space for potential included arguments can shrink drastically, especially in AFs that contain many ar-

guments that are sparsely connected by attacks. The removal of irrelevant argument and attacks is assumed to be especially effective when applying it to the choice rules for *att_to_remove*, since the size of search space should be the biggest hinderance given the exponential nature of the search problem.

6.2.4 Implementing ModAttDel

Provided the implementation of *AttDelScore*, the implementation of *ModAttDelScore* requires minor adjustments. To calculate *AttDelDenyScore*, the enforcement constraint of *AttDelScore* needs to be flipped

$$:- \text{grounded}(X), \text{enf}(X). \quad (16)$$

The constraint (16) enforces that it is impossible for the enforced argument X to be grounded.

This logic cannot be applied to the implementation with strongly admissible sets.

$$:- \text{strongly_admissible}(X), \text{enf}(X). \quad (17)$$

Since ASP operates on answer sets, instead of finding all strongly admissible sets and evaluating the presence of the enforced argument in one of them, its answer sets only find one strongly admissible set. Since the empty set is strongly admissible, the minimal number of attack deletions to enforce the enforced arguments to not be in a strongly admissible set is always 0. Proving non-existence of the argument in any strongly admissible set requires the calculation of all strongly admissible sets which equates to calculating the grounded extension. Since we already have an efficient way of calculating the grounded extension, the approach of using strongly admissible sets is ineffective for negative *ModAttDelScores*. The strongly admissible implementation approach is therefore used for positive *ModAttDelScores*, with negative scores being calculated by the grounded approach.

6.2.5 Performance

Given the value that the semantics provides, the second important factor for the application of the semantics is its performance. Due to the NP-hard nature of the semantics, its implementations are likely to become exponentially more complex with larger AFs until a point where it loses its viability. In this section, we investigate when that point is reached. To ensure meaningful benchmark results, we are using a diverse set of AFs from the 2025 ICCMA competition [2]. Instead of calculating *ModAttDel* for all arguments, we target three arguments per AF of varying supposed difficulty of extension enforcement. The first argument is an argument that is part of the grounded extension. This ensures the inclusion of the calculation of *ModAttDelScore*. The second argument is an argument that is part of a complete extension but is not part of the grounded extension. Such an argument has an existent support structure and an AF may not need many alterations for it to also be generally accepted. The third targeted argument is an argument that is

rejected by the complete semantics. An additional goal in this section is assessing if the problems of the approach using strongly admissible sets translate into worse performance.

Previous work by Niskanen et al. [18] demonstrates that MaxSAT implementations vastly outperform ASP solutions for extension enforcement. Because this performance gap is already established, the goal of this section is not to achieve state-of-the-art runtime. Instead, we use ASP to isolate and analyze the performance behavior of different logical formulations. By this, we provide baseline data to inform the design of future solvers.

The benchmark is performed on the dataset's 200 AFs with the smallest number of attacks, ranging from 111 to 74998 attacks. Each argument receives a timeout threshold of 60 seconds. An instance is solved if all selected arguments are solved. We analyze the average solve time in the benchmark for the selected arguments. We perform the pruned and unpruned approach of the strongly admissible and grounded implementation for *ModAttDelScore* respectively. The benchmark is run on a Windows machine using a 6-core processor with a 3.6 GHz clock speed able to boost to 4.2 GHz and 32 GB of memory. Clingo is configured to utilize all available CPU-cores.

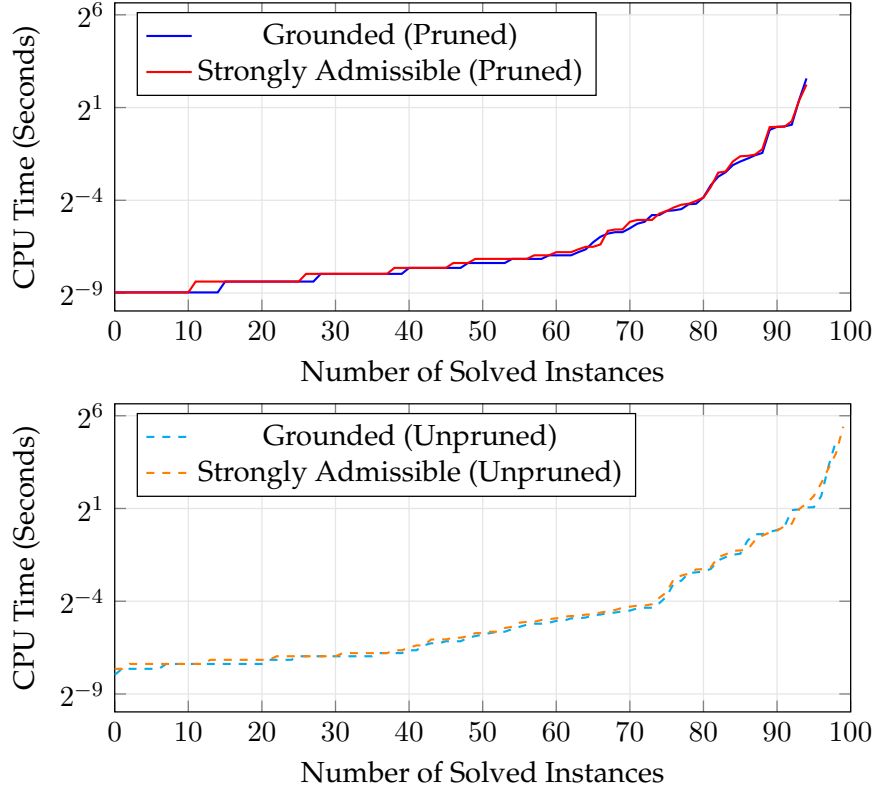


Figure 19: Comparison of *ModAttDel* calculation using grounded extension and strongly admissible approach.

Comparing the computational performances of the approach using the grounded extension and the approach using strongly admissible sets, as seen in Figure 19, shows no significant difference in performance. The presumed increase in computational load due to the nested search problem for the strongly admissible approach did not occur in this benchmark. Given the number of solved instances for both pruned approaches is 99, we can assume nearly equal performance between both approaches.

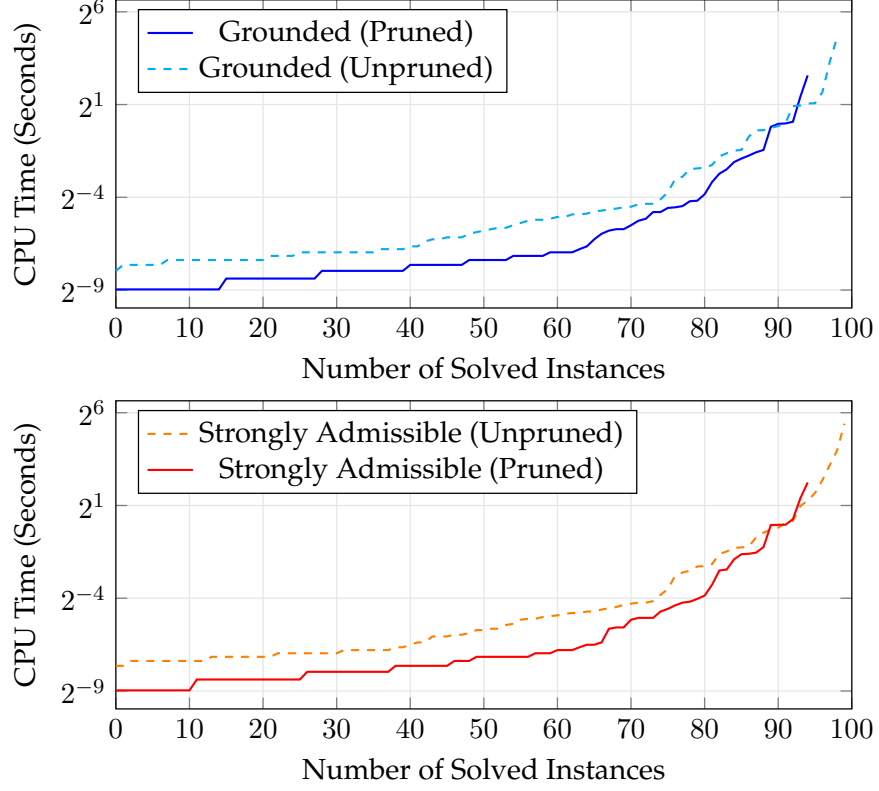


Figure 20: Pruning Performance for Grounded and Strongly Admissible Approach.

As seen in Figure 20, pruning the search space for both approaches leads to decreased computational load initially. However, given the pruned versions only solved 95 instances before timing out, while the unpruned variants solved 99 (grounded) and 100 (strongly admissible), we cannot assume any significant performance increase by pruning, with the opposite being the case in this benchmark.

The results demonstrate performance parity between the grounded and strongly admissible approaches. The higher theoretical complexity of the SA logic did not result in a significant runtime penalty, indicating that the primary bottleneck is the inherent complexity of the semantics. Both methods remain viable computational baselines for extension enforcement.

7 Conclusion

This thesis aimed to introduce a novel ranking-based semantics that ranks arguments in an AF by a distance to the grounded extension. Such a semantics captures the intuition of universally accepted arguments, by utilizing the grounded semantics as the key reference point to rank by. To facilitate such a ranking, we have

introduced a scoring system *AttDelScore* which represents the minimum number of attack deletions to enforce a targeted argument in the grounded extension. The score utilizes non-strict extension enforcement, which is an NP-hard search-optimization problem. By comparing *AttDelScores* between arguments, we built a preorder θ^{AttDel} , referred to as AttDel, our Attack Deletion ranking-based semantics. Due to its intuitive and comprehensive nature, AttDel is able to map realistic argumentation scenarios, as shown in the introductory chapter, and perform a ranking closely matching human intuition, while being quite comprehensive in its way of working.

To assess the structural strengths and weaknesses of AttDel, we have analyzed how it behaves in reference to common key principles. This initial analysis has shown that AttDel belongs to the global category of ranking-based semantics, where, in contrast to local semantics, the direct neighborhood does not have any inherent value in the ranking, and any argument with a large distance to the targeted argument can have equal impact to its ranking as a close neighbor. Local ranking-based semantics, like SAF or Categorizer often comply principles regarding the direct neighborhood and direct attackers like cardinality precedence, quality precedence or counter transitivity, which specifically target direct attackers [6]. Global ranking-based semantics like Tuples*, M&T have a global scope and cannot discriminate based on distance to the target argument [6]. Since a grounded extension can massively change with simple changes in the attack structure regardless of the distance to the target argument, AttDel belongs in the global category. These ranking-based semantics comply with Attack vs Full Defense, as their ranking scope spans along full attacking and defense branches and a more accurate assessment of defeat can be made.

The principle analysis has shown a unique behavioral profile of AttDel, being compatible with the grounded extension, which in the set of compared semantics only the grounded semantics is capable of, while coming with the significant drawback of being a binary semantics that lacks nuance.

The principle analysis has also shown that AttDel does not comply with void precedence as it values arguments accepted by the grounded semantics equally regardless of whether it is attacked. This finding demonstrates a weakness of AttDel, that its granular method of ranking arguments only applies to arguments outside of the grounded extension. In an attempt of solving this problem, we have introduced a modified version of AttDel, *ModAttDel*.

ModAttDel preserves the ranking of arguments outside of the grounded extension established by AttDel. For accepted arguments, it introduces a new, structurally similar ranking approach aimed at finding tipping points at which an argument becomes rejected by the grounded extension by attack deletions. While it rewards weak attack structures for non-grounded arguments, equally to AttDel, it also punishes weak defense structures for grounded arguments. Analyzing the compliance of the established principles by *ModAttDel* has shown no difference to AttDel, except the compliance with void precedence. Due to this, *ModAttDel* complies with

all established baseline principles for ranking-based semantics, further underlining the value of the semantics.

In the final section, we have shown approaches of implementing AttDel and ModAttDel. Due to its NP-hardness, imperative, deterministic approaches struggle with the exponential complexity making them unviable for the problem. As the literature has shown, extension enforcement can be efficiently implemented using specialized solvers. As a proof of concept we have implemented AttDel and ModAttDel using answer set programming (ASP), using the Clingo grounder and solver to successfully compute *AttDelScore* and *ModAttDelScore* for small AFs in reasonable runtime. In an attempt of runtime optimization within the bounds of Clingo, we have created different approaches of implementation. The approach using the calculation of the grounded extension most closely matches the formal definition of AttDel, while the approach using strongly admissible sets attempts to reduce computational overhead by only calculating a subset of the grounded extension. We have also experimented with preprocessing the search space by pruning arguments and attacks that logically cannot be part of the resulting set of attack deletions. Since all approaches and optimizations performed relatively equal, we can assume that the advanced solving logic of Clingo does significant optimization by itself, eliminating the need for additional manual tweaking.

7.1 Limitations

A notable theoretical limitation of both AttDel and ModAttDel is their non-compliance with a significant number of established principles for ranking-based semantics. As the comparison of properties illustrated, while they satisfy fundamental baselines like Abstraction and Independence, they fail to comply with numerous standard local and global principles, including Cardinality Precedence, Quality Precedence, Defense Precedence, and Counter-Transitivity. This widespread non-compliance stems directly from their core design: by relying strictly on extension enforcement and the grounded extension, the semantics prioritize the overarching acceptability status of arguments over granular topological features, such as direct attacker counts or specific defense branch structures or lengths. Consequently, in applications where compliance with these specific structural properties is strictly required, AttDel and ModAttDel may not be suitable choices. Given the performance analysis has not found significant differences between approaches, we have not been able to find meaningful optimizations within the bounds of Clingo. Further analysis would be required to assess this. Within the constraints of the analysis, of the 200 least complex AFs from the test set, at least one half of the attempts timed out yielding no result. The Clingo approach is therefore not a solution that can be used in productive environments with comparable AFs to the test set. However, as Niskanen et al. [18] had shown, MaxSAT implementations of extension enforcement perform much better, showing a bigger likelihood of the new semantics being viable in practical applications.

The concept of the attack deletion semantics also poses limitations. The semantics is not sensitive to double standards, which can be shown in the following example:

Example 5 (Double Standard). We introduce a scenario where a vehicle system has to decide which services to keep running while low on battery.

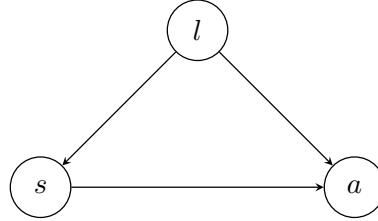


Figure 21: AF symbolizing a car system behavior on low battery.

The AF shown in Figure 21 shows three arguments:

- l : The battery is low and resources must be limited.
- s : High Priority: The car safety system must stay enabled to preserve safety of the passenger.
- a : Low Priority: The air conditioning must stay enabled so the passenger is comfortable.

Paradoxically, AttDel ranks s and a equally ($AttDelScore = 1$), as the deletion of the respective direct attack from l causes both arguments to be part of the grounded extension. Following the example, the battery being low justifies the preservation of air condition over the clearly more important safety systems. At the same time, the battery being low is ignored by the air conditioning itself to justify staying on. This is a double standard and in this abstract example, AttDel is a bad choice of ranking. A semantics that is more sensitive to double standards, while keeping a similar approach, would be an ArgDel semantics that enforces arguments in the grounded extension with a minimum number of arguments deleted. In this example, arguments l and s would require deletion to make a part of the grounded extension, while only l would require deletion to make s part of the grounded extension. Such a semantics would therefore result in the expected ranking, which AttDel fails to.

7.2 Future Work

As this thesis has shown, the approach of attack deletion to enforce a semantics and establish a preorder offers high potential for intuitive and comprehensive ranking. This concept can be easily extended to other extension semantics depending on the users definition of *accepted*. The tendency of exponential runtime increases warrants

the deployment of more advanced solvers for the calculation of AttDel and to investigate other targeted optimization techniques for the implementation. Very similar implementations using different MaxSat frameworks already exist [18], showing high performance compared to Clingo, and require only minor changes and the inclusion of *AttDelDenyScore* to successfully map ModAttDel.

Another potential topic of research is investigating the deletion of arguments for extension enforcement and ranking by the minimum set of deletions. AttDel works on accepting and minimizing the consequences of certain attacks, the acceptance of consequences could also be carried forward to all other consequences that an argument poses. This more invasive approach, leading to more jarring changes may be more logically consistent as double standards of attack acceptance are prevented.

References

- [1] Leila Amgoud and Jonathan Ben-Naim. Ranking-based semantics for argumentation frameworks. In Weiru Liu, V. S. Subrahmanian, and Jef Wijsen, editors, *Scalable Uncertainty Management - 7th International Conference, SUM 2013, Washington, DC, USA, September 16-18, 2013. Proceedings*, Lecture Notes in Computer Science, pages 134–147. Springer, 2013.
- [2] Iosif Apostolakis, Andrei Popescu, and Johannes P. Wallner. Sixth international competition on computational models of argumentation: Preliminary report, 2025.
- [3] Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.
- [4] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235, 2001.
- [5] Lydia Blümel and Matthias Thimm. A ranking semantics for abstract argumentation based on serialisability. In Francesca Toni, Sylwia Polberg, Richard Booth, Martin Caminada, and Hiroyuki Kido, editors, *Computational Models of Argument - Proceedings of COMMA 2022, Cardiff, Wales, UK, 14-16 September 2022*, Frontiers in Artificial Intelligence and Applications, pages 104–115. IOS Press, 2022.
- [6] Elise Bonzon, Jérôme Delobelle, Sébastien Konieczny, and Nicolas Maudet. A comparative study of ranking-based semantics for abstract argumentation. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 914–920. AAAI Press, 2016.
- [7] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Graduality in argumentation. *Journal of Artificial Intelligence Research*, 23:245–297, 2005.
- [8] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. Extension enforcement in abstract argumentation as an optimization problem. In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2876–2882. AAAI Press, 2015.
- [9] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.
- [10] Wolfgang Dvorák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. *IfCoLog Journal of Logics and their Applications*, 4(8), 2017.

- [11] Wolfgang Dvorák, Anna Rapberger, Johannes P. Wallner, and Stefan Woltran. ASPARTIX-V19 - an answer-set programming based system for abstract argumentation. In Andreas Herzig and Juha Kontinen, editors, *Foundations of Information and Knowledge Systems - 11th International Symposium, FoIKS 2020, Dortmund, Germany, February 17-21, 2020, Proceedings*, Lecture Notes in Computer Science, pages 79–89. Springer, 2020.
- [12] Wolfgang Dvorák and Johannes P. Wallner. Computing strongly admissible sets. In Henry Prakken, Stefano Bistarelli, Francesco Santini, and Carlo Taticchi, editors, *Computational Models of Argument - Proceedings of COMMA 2020, Perugia, Italy, September 4-11, 2020*, Frontiers in Artificial Intelligence and Applications, pages 179–190. IOS Press, 2020.
- [13] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [14] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.
- [15] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming*, 19(1):27–82, 2019.
- [16] João Leite and João G. Martins. Social abstract argumentation. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2287–2292. IJCAI/AAAI, 2011.
- [17] Paul-Amaury Matt and Francesca Toni. A game-theoretic measure of argument strength for abstract argumentation. In Steffen Hölldobler, Carsten Lutz, and Heinrich Wansing, editors, *Logics in Artificial Intelligence, 11th European Conference, JELIA 2008, Dresden, Germany, September 28 - October 1, 2008. Proceedings*, Lecture Notes in Computer Science, pages 285–297. Springer, 2008.
- [18] Andreas Niskanen, Johannes P. Wallner, and Matti Järvisalo. Extension enforcement under grounded semantics in abstract argumentation. In Michael Thielscher, Francesca Toni, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018*, pages 178–183. AAAI Press, 2018.
- [19] Johannes P. Wallner, Andreas Niskanen, and Matti Järvisalo. Complexity results and algorithms for extension enforcement in abstract argumentation. *Journal of Artificial Intelligence Research*, 60:1–40, 2017.

- [20] Ulla Wege. Aggregierte rangbasierte Semantiken für abstrakte Argumentation. Master's thesis, FernUniversität in Hagen, Faculty of Mathematics and Computer Science, Artificial Intelligence Group, 2024.