Defining Defense in Abstract Argumentation from Scratch - A Generalizing Approach

Masterarbeit

zur Erlangung des Hochschulgrades

Master of Arts (M.A.)

vorgelegt von

Lydia Blümel

(geboren am 11.07.1997 in Elsterwerda)

betreut von Dr. Markus Ulbricht (Institut für Informatik)

Tag der Einreichung: 31. 01. 2022

Matrikelnummer: 3758189

Erstgutachten: Dr. Markus Ulbricht

Zweitgutachten: Prof. Ingolf Max

Contents

Chapter 1

Introduction

Abstract Argumentation has been a highly active field of AI-research for decades and it is of particular relevance in the context of multi-agent systems [Carrera and Iglesias, 2015]. Its applications range from formal argumentation in philosophy [Prakken, 2011] over legal reasoning [Calegari et al., 2019] up to decision making in medicine [Fox et al., 2007, Bromuri and Morge, 2013]. A decisive work for this trend was the article "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games" by Dung in 1995.[1] He took a set of arguments, for instance in a political discussion, and analyzed them purely based on their conflicts with each other. The interpretation of such an argumentation framework was then formalized by so called argumentation semantics. During the three decades between then and now his concept is being constantly expanded and many researchers in the field of argumentation have contributed by developing their own argumentation semantics [Baroni et al., 2011]. A relatively young category among those semantics are so called weak semantics. They were designed for the purpose of accepting more arguments than classic semantics in certain problematic cases.

Imagine for example three companies A, B, C, all of the same sector. Suppose A and B are equally large and successful, while C is comparatively small. Naturally A and B each try to dominate the market by taking over one of the other two companies. Now A cannot buy B because B's market value is too high and vice versa. Although C is small and cannot resist being taken over by A directly, A cannot buy C either because it would lead in a bidding competition with B. That would mean either loosing C to B or potentially getting into a financial crisis by the take over. One can say A and B successfully defend themselves with their own power against being taken over by each other. This much can be adequately modeled

[1] [Dung, 1995]

by classic Dung-style semantics. But Company C survives for a different reason, namely the deadlock between A and B, a fact classic semantics cannot account for. What frequently seems to happen now is that researchers are confronted with a case like the above example and conclude: "Dungs semantics does not work correctly here." Immediately after reaching this conclusion they are eager to propose something that works. This has lead to a surprising variety of solutions [Bodanza and Tohmé, 2009, Dondio and Longo, 2018, Baumann et al., 2020b, Dauphin et al., 2020] for this and/or related problems. We are now in a similar situation with weak semantics like Baroni&Giacomin were 2007 with general semantics when they said "The increasing variety of semantics proposed in the context of Dung's theory of argumentation makes more and more inadequate the example-based approach commonly adopted for evaluating and comparing different semantics." in the abstract of [Baroni and Giacomin, 2007]. The principle-based analysis proposed by them has become the standard procedure for comparing different semantics in various categories. This kind of analysis was conducted for weak semantics, too, in [Baumann et al., 2020a, Dauphin et al., 2020, Dondio and Longo, 2021]. Thanks to their work the overall performance of the various weak semantics is already well understood, but the same cannot be said about their defense behavior.

Our first step towards a better understanding of this is extracting the resp. defense concepts of a selected number of weak semantics. In order to do this an introduction to Dung-style argumentation and to existing weak semantics is given in Chapter 2, Previous Work. As SCC-semantics are already a category of their own, we decided to exclude semantics based on SCC-recursiveness like the ub-semantics in [Dondio and Longo, 2021] or the qualified semantics in [Dauphin et al., 2020] from the scope of this work. The reader who is familiar with Dung-style argumentation frameworks may want to forgo Section 2.1 and 2.2 but is advised to catch up with the semantics introduced in Section 2.3. The aim of Chapter 3 is to develop a general notion of defense for abstract argumentation that allows us to categorize those weak semantics according to their underlying defense philosophy. We demonstrate the new concept by applying it to the semantics from Section 2.3 and by defining a new weak semantics based on it. The remaining chapters are structured as follows. Chapter 4 is dedicated to the comparison and analysis of the aforementioned semantics under the new notion. In Chapter 5 we reexamine the motivation behind weak argumentation and propose a number of properties weak semantics should satisfy. In Chapter 6 we investigate the usability of the introduced semantics for the ASPIC framework. We close with a short discussion of the results and related as well as future work in Chapter 7.

Chapter 2

Previous Work

This chapter starts with a formal definition of Dung-style Argumentation Frameworks and argumentation semantics which will serve as the formal background for the concepts and results presented in this work. Next, we will first introduce classic Dung-style-semantics [Dung, 1995], followed by three different approaches to weak semantics in chronological order- the works of Bodanza&Tohmé on the problems of self-attackers and odd cycles [Bodanza and Tohmé, 2009, Bodanza et al., 2014], the undecidedness blocking concept of Dondio&Longo for handling undecided attackers in label-based semantics [Dondio and Longo, 2018, Dondio and Longo, 2021] and the recently publicized recursive semantics of Baumann, Brewka and Ulbricht [Baumann et al., 2020b] whose aim is to preserve a concept of defense with their approach. We present only three new results in this chapter - an extension-based version of the weakly complete semantics from [Dondio and Longo, 2021], a consequence of modularization we call *persisting non-admissibility* and a proof concerning the directionality of the $\exists$-semantics from [Dauphin et al., 2021]. All other definitions and results are taken from the resp. literature.

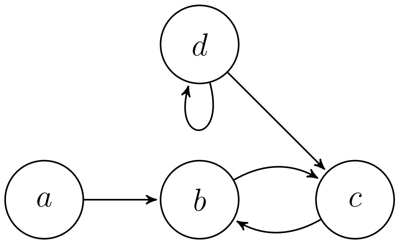
2.1 Basic ideas of Dungs Abstract Argumentation Theory

We will follow the introductions from [Baumann et al., 2020b] and [Baroni et al., 2011]. Let U_{arg} be an infinite set - the *universe of arguments*.

Definition 2.1 (AF). An *Argumentation Framework(AF)* $F = (A, R)$ is a tuple of a finite set of arguments $A \subset_f U_{arg}$ and a binary relation $R \subseteq A \times A$ on A, which is called the *attack relation*.
Let U_F be the set of all finite argumentation frameworks over U_{arg}.

A Dung-style argumentation framework has two components, the arguments and the attacks between the arguments. Both have no inner structure which is on the one hand a virtue, on the other a weakness. The simplicity of the system allows a wide range of applications and is very close to human argumentation in practice. The downside is that there is very little information in an AF to work with. For example there is no form of direct defense meaning attacks cannot be blocked. An AF $F = (A, R)$ is usually represented by a directed graph with A as its set of vertices and R as its set of edges.

Example 2.2. The AF $F = (A, R)$ with $A = \{a, b, c, d\}$ and
$R = \{(a, b); (b, c), (c, b), (d, c), (d, d)\}$ is represented by the following digraph.



In general there are no limitations to the attack relation whatsoever, an argument may attack itself, like d in Example 2.2, there may be no attacks at all and so on. Note that there is at most one attack from the same argument a to the same argument b, since there is only one pair $(a, b) \in A \times A$, and that the attack (c, b) is distinct from (b, c), since R is not symmetric in general.The symbol "$\rightarrow$" will be used as a shorthand for attacks as follows.

Definition 2.3. Let $F = (A, R)$ be an AF. An argument $a \in A$ attacks another argument $b \in A$, written as $a \rightarrow b$ iff $(a, b) \in R$. An argument a is unattacked iff no attacks on a exist that is if $(b, a) \notin R$ for every argument $b \in A$ (including a itself).
A set of arguments $E \subseteq A$ attacks another set $D \subseteq A$, in short $E \rightarrow D$ iff there exist arguments $e \in E$, $d \in D$ such that $e \rightarrow d$. We say E is unattacked iff E is not attacked by any arguments $a \in A \setminus E$ not belonging to E.[1]

Example 2.4. In Example 2.2 a attacks b and $\{a, b\} \rightarrow c$.[2] Note the difference between the unattacked argument a and the unattacked argument set[3] $\{d\}$.

As in the above definition, small letters will denote arguments and capital letters will denote both sets of arguments and AFs from here on. The capital letter R is reserved for attack relations.

[1]There may still be attacks among the arguments of E.
[2]$E \rightarrow a$ is a shorthand for $E \rightarrow \{a\}$, $a \rightarrow E$ vice versa.
[3]short for "set of arguments"

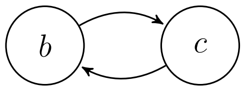
The following definitions become useful when speaking about attacks and, more importantly, defense between argument sets instead of single arguments.

Definition 2.5. Let $F = (A, R)$ be an AF and $E \subseteq A$. The set of all attackers of E is given by $E^- := \{a \in A \mid a \to E\}$ and analogously $E^+ := \{a \in A \mid E \to a\}$ is the set of all arguments attacked by E. We define the *range* of E to be the argument set $E^\oplus := E \cup E^+$.

Definition 2.6 (Sub-AF)**.** Let $F = (A, R)$ be an AF and $E \subseteq A$. Then the *restriction* of F on E is the AF $F{\downarrow_E} = (E, R \cap (E \times E))$, the Sub-AF of F with argument set E.

When the AF we refer to is clear, an argument set and the restriction of an AF to that argument set will be denoted by the same variable (a capital letter, e.g. E).

Example 2.7. Example 2.2 continued. The set of attackers of $\{b, c\}$ is $\{b, c\}^- = \{a, b, c, d\}$, $\{b, c\}^+ = \{b, c\}$ itself, which is also the range $\{b, c\}^\oplus = \{b, c\}$. The restriction of F on $\{b, c\}$, $F{\downarrow_{\{b,c\}}}$ is the AF below.



A reader familiar with graph theory may have noticed that the concepts introduced so far have counterparts in the field of digraph study, e.g. Sub-AFs are subgraphs, basically. While Dung-style Formal Argumentation makes good use of the rich knowledge on digraphs in existence its focus lies elsewhere. The core of this research field are the so called argumentation semantics. They are used to formalize the process of determining potential "winners" of the conflict represented by an AF. Our introduction to semantics here is based on [Baroni et al., 2011].

Definition 2.8 (extension-based semantics)**.** An extension-based argumentation semantics is a mapping $\varsigma_{ext} : U_F \to 2^{2^{U_{arg}}}$ returning for every AF $F = (A, R)$ a set of extensions $\varsigma_{ext}(F) \subseteq 2^A$. A ς-extension of F is an argument set $E \in \varsigma_{ext}(F)$.

Although the combination of arguments with an attack relation is not exactly the prototype of what one would call a logical syntax, an argumentation semantics is exactly what the name implies - a semantics. It computes the possible values of an argument based on the conflicts it engages in. The similarities to formal semantics for e.g. propositional calculus become even more obvious when the argumentation semantics takes the form of sets of labelings.

Definition 2.9 (label-based semantics)**.** Let $F = (A, R)$ be an AF. A Δ-labeling[4] on F is a mapping $lab : A \to \{in, out, undec\}$ which assigns to each argument of F one of the three states *in*, *out* or *undec*.
A label-based semantics is a mapping $\varsigma_{lab} : U_F \to 2^{\{lab \mid lab \colon U_{arg} \to \{in,out,undec\}\}}$ returning for each AF $F = (A, R)$ a set $\varsigma_{lab}(F)$ of labelings on F.[5]

A label-based semantics ς defines which labelings interpret the conflicts in a given AF "correctly", much like truth functions determine which variable assignments to truth-values are legit for a set of formulas in propositional calculus. By contrast extension-based semantics are not that function-oriented and more concerned with an intuitive understanding of when and how arguments work together. Focusing on the behavior of groups of arguments comes with many advantages, especially when studying the various intuitions about defense. For this reason we will focus on extension-based semantics in this work. Most semantics can be defined both as an extension-based and as a label-based semantics,[6] in these cases ς_{ext} denotes the set of extensions and ς_{lab} the set of labelings w.r.t to the same semantics ς. We will use ς to refer to an argumentation semantics in general and the reader may assume it is extension-based where no further specifications are given.

Some semantics take the form of elaborate algorithms, like the SCC-semantics in [Dondio and Longo, 2021]. But most of them come in the form of a set of rules that an extension resp. a labeling has to satisfy. The following definition introduces a basic semantics that is often not even considered a semantics in its own right but a property of any rational semantics.

---

[4]The only kind of labeling considered in this work, for a more general introduction to labelings on AFs see [Baroni et al., 2011]

[5]To be precise $\varsigma_{lab}(F) \subseteq \{lab \mid lab :\ U_{arg} \to \{in, out, undec\}\}$ is defined as a set of labelings on all arguments but we only consider the restrictions of these labelings $\{lab\restriction_A) \mid lab \in \varsigma_{lab}(F)\}$ when talking about ς-labelings on F. To simplify this we redefine $\varsigma_{lab}(F) \coloneqq \{lab\restriction_A) \mid lab \in \varsigma_{lab}(F)\}$ to denote the set of the restrictions on F of labelings assigned by ς to F.

[6]extension-based can be translated to label-based without problems but the reverse is only straightforward for semantics satisfying rejection, see [Baroni et al., 2011]

Definition 2.10 (conflictfree semantics)**.** The cf-semantics assigns to each AF $F = (A, R)$ the set $cf(F) = \{E \subseteq A \mid E\ conflictfree\}$ of all conflictfree argument sets in F. An argument set $E \subseteq A$ is conflictfree iff there are no attacks among the arguments of E that is for any arguments $a, b \in E$ we have $(a, b) \notin R$.
An extension-based semantics ς has the property of being *conflictfree* iff $\varsigma(F) \subseteq cf(F)$.

Example 2.11. Example 2.2 continued. The set of all conflictfree extensions of F is $cf(F) = \{\emptyset, \{a\}; \{b\}; \{c\}; \{a, c\}\}$.

The main reason why we introduce the conflictfree semantics as a semantics in its own right is that it is a good reference for a semantics with no defense at all. And since conflictfreeness is widely respected as a necessary characteristic of a good argumentation semantics, the cf-semantics will serve us as a lower bound for all attempts to weaken classic defense. The cf-semantics can be formulated as a label-based semantics too. Like for the extension version, only one rule is needed.

Definition 2.12 (cf-labeling)**.** Let $F = (A, R)$ be an AF. $lab : A \to \{in, out, undec\}$ is a *cf-labeling*, $lab \in cf_{lab}(F)$ iff for any argument $a \in A$ it holds that $lab(a) = out$ iff an attacker $b \in \{a\}^-$ exists with $lab(b) = in$.
A label-based semantics ς_{lab} satisfies the *rejection property* iff $\varsigma_{lab}(F) \subseteq cf_{lab}(F)$.[7]

Note that despite the labelings being three-valued, there is a one-on-one mapping between extensions and labelings with this labeling rule because the set of out-labeled arguments is determined by the set of in-labeled arguments and the undecided arguments end up being the rest. This is the so-called rejection property and any labeling-semantics satisfying it can be translated to an extension-based semantics without information loss. We will now give a short demonstration in which sense these two definitions describe the same semantics.

Proposition 2.13. *Let $F = (A, R)$ be an AF. Then $cf_{ext}(F) = \{in(lab) \mid lab \in cf_{lab}(F)\}$ that is for each cf-extension E exists a cf-labeling lab such that both $E = \{a \in A \mid lab(a) = in\}$ and $E^+ = \{a \in A \mid lab(a) = out\}$ hold and vice versa.*

Example 2.14. Example 2.2 continued. A cf-labeling on F can be constructed as follows: Suppose we set $lab(b) = in$ then $lab(c) = out$. Since a has no attackers $lab(a) \neq out$ and since it attacks b, which is labeled in, $lab(a) \neq in$ so

[7] Our definition of cf-labelings is a special case, it is possible to define conflictfree labelings as those where $in(lab)$ is conflictfree(see [Baroni et al., 2011] for details), however, we decided to give an example of a bijective relationship between a labeling-based and an extension-based semantics here

$lab(a) = undec$. Because d attacks itself labeling it *in* would lead to a contradiction and since no other in-labeled attacker is available $lab(d) = undec$. This labeling coincides with the cf-extension $\{b\}$, for the construction of the other three cf-labelings the same method can be applied.

Proof. Let $F = (A, R)$ be an AF and let $E \in cf(F)$. Then $lab_E : A \to \{in, out, undec\}$ with $lab_E(a) = in$ if $a \in E$, $lab_E(a) = out$ if $a \in E^+$ and $lab_E(a) = undec$ else for any argument $a \in A$ satisfies the condition of Def. 2.12. Now let $lab \in cf_{lab}(F)$ be a cf-labeling, then $in(lab)$ is conflictfree, because any argument b attacked by an argument $a \in in(lab)$ is labeled *out* and the set of all out-labeled arguments is exactly $in(lab)^+$. □

The idea behind argumentation semantics is that a semantics evaluates a given AF to decide which arguments can be accepted together by a rational agent. Those groups of acceptable arguments are the extensions or, in case of labeling-semantics, the in-labeled arguments of a certain labeling. But that alone is not enough for an agent to decide e.g. which arguments should be added to a knowledge base, since different extensions can be in conflict with each other and there is no preference ranking among them. The question if a single argument is accepted by a certain semantics is therefore commonly answered with one of the two following options.

Definition 2.15 (acceptability)**.** Let $F = (A, R)$ be an AF. An argument $a \in A$ is *credulously accepted* w.r.t. a label-based semantics ς iff some labeling $lab \in \varsigma(F)$ exists such that $lab(a) = in$. a is credulously accepted by an extension-based semantics ς, denoted by $a \in_{ext} \varsigma(F)$, iff $a \in \bigcup_{E \in \varsigma(F)} E$ that is if a is an element of some ς-extension E of F.
a is *skeptically accepted* w.r.t. a label-based semantics ς iff $lab(a) = in$ for all $lab \in \varsigma(F)$, w.r.t an extension-based semantics ς, denoted by $a \in_{sk} \varsigma(F)$, iff $a \in \bigcap_{E \in \varsigma(F)} E$ that is if $a \in E$ for all $E \in \varsigma(F)$.

Example 2.16. For the AF of Example 2.2 the arguments a, b, c are credulously accepted by the cf-semantics. Since the empty set is a cf-extension, no argument is skeptically accepted.

Because skeptical acceptance ensures conflictfreeness among the accepted arguments (if a conflictfree semantics is used) it is the preferred form of acceptance for applications of formal argumentation. With this comes the problem that for two extensions E, E' with $E \subset E'$ only the arguments of E can be skeptically accepted. The common solution for this is limiting a semantics to its maximal extensions, in case of cf-semantics this restriction is called naive semantics.

Definition 2.17 (naive semantics)**.** Let $F = (A, R)$ be an AF. The set of all naive extensions of F is $na(F) = \{E \in cf(F) \mid \forall D \in cf(F) : E \subseteq D \Rightarrow E = D\}$ the set of all maximal conflictfree argument sets in F.

To give an informal example of this semantics, in Example 2.2 the cf-extension $\{a\}$ is not naive, because its superset $\{a, c\}$ is a cf-extension too. Therefore $na(F) = \{\{a, c\}; \{b\}\}$. The set of skeptically accepted arguments is still empty in this special case, but suppose an additional argument existed that is not in conflict with a, b, c, d or itself, then any naive extension had to contain it and it would be accepted skeptically. The following proposition shows that limiting cf-semantics to its maximal extensions does not impact credulous acceptance, so no accepted argument is "lost" by limiting ourselves to maximal extensions.

Proposition 2.18. *Let $F = (A, R)$ be an AF. An argument $a \in A$ is creduously accepted by the cf-semantics, $a \in_{ext} cf(F)$ iff $a \in_{ext} na(F)$ that is a is creduously accepted by naive semantics.*

Proof. $a \in_{ext} cf(F)$ means there exists some $E \in cf(F)$ such that $a \in E$. Now either E is already maximal or some $D \in na(F)$, $D \supset E$ exists, then $a \in D$. So either way $a \in_{ext} na(F)$. □

2.2 Classic defense and Dung-style semantics

In this section we will recapitulate the core definitions and results from [Dung, 1995]. There Dung introduces five semantics which are based on two simple principles: A set of arguments is acceptable if it has no inner conflicts among its members (1) and defends itself against all attackers from the outside (2). Conflict-freeness (1) was already introduced in the previous section. What Dung precisely means by defense is stated in the following definition.

Definition 2.19 (c-defense)**.** Let $F = (A, R)$ be an AF, $E \subseteq A$ and $a \in A$. a is classically(c) defended by E iff for any attacker $b \in \{a\}^-$ an $e \in E$ exists such that $e \rightarrow b$.
The classic Defense-Operator $\Gamma : 2^A \rightarrow 2^A$ maps each argument set E in F to the set $\Gamma(E) := \{a \in A \mid E\ c-defends\ a\}$ of arguments it classically defends.

Defense by attack, to put it simply. That makes sense in a formalism where the only relation is an attack relation. The notable achievement here lies in describing this concept with an operator. Now c-defense can be characterized in terms of operator properties like the following.

Proposition 2.20 (monotonicity)**.** *Let $F = (A, R)$ be an AF. Then Γ is monotonic on 2^A that is for any $E, D \subseteq A$ if $E \subseteq D$ then $\Gamma(E) \subseteq \Gamma(D)$.*

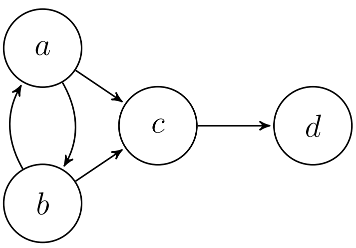
Proof. Let $a \in A$ be an argument c-defended by E, then any attacker $b \in \{a\}^-$ is attacked by some $e \in E$. Since $E \subseteq D$, we have $e \in D$ so a is c-defended by D. □

Together with conflictfreeness we get the c-admissible semantics.

Definition 2.21 (c-admissibility)**.** The classic Dung-style admissible semantics $ad^c : U_F \to 2^{2^{U_{arg}}}$ is defined for any AF $F = (A, R)$ as follows: Let $E \subseteq A$ then $E \in ad^c(F)$ iff E is conflictfree and $E \subseteq \Gamma(E)$ (E c-defends at least itself).

Note that the empty set is always c-admissible because it has no arguments that could be attacked. We will now demonstrate the relationship between c-defense and c-admissibility with an example.

Example 2.22.



In the AF above $\{a\}$ c-defends itself and d, so both $\{a\}$ and $\{a, d\}$ are c-admissible, while $\{d\}$ is not, because it cannot defend itself against c. In contrast the set $\{a, b\}$ does c-defend itself, but is not c-admissible because it is not conflictfree. The set of all c-admissible extensions is $ad^c(F) = \{\emptyset; \{a\}; \{a, d\}; \{b\}; \{b, d\}\}$

It turns out conflictfreeness is preserved under the addition of c-defended arguments to a c-admissible extension, for example $\{a\}$ is conflictfree and so is $\{a, d\}$. This useful property of c-defense is described by the following Lemma which is considered one of the central results of [Dung, 1995].

Theorem 2.23 (Fundamental Lemma)**.** *Let $F = (A, R)$ be an AF and $E \subseteq A$ a c-admissible extension of F ($E \in ad^c(F)$) and let $a, b \in \Gamma(E)$ be arguments c-defended by E. Then $E \cup \{a\} \in ad^c(F)$ and b is c-defended by $E \cup \{a\}$.*

Proof. After showing that $E \cup \{a\}$ is conflictfree the rest follows from Prop. 2.20. Suppose $E \cup \{a\}$ is not conflictfree, then because E is conflictfree some $e \in E$ exists such that either $e \to a$ (1) or $a \to e$ (2). If (1) then because a is c-defended by E some $d \in E$ must exist that attacks e. This contradicts the presupposed conflictfreeness of E. If (2) e is c-defended by E against a because $E \subseteq \Gamma(E)$. Therefore some $d \in E$ exists which attacks a. This results in a repetition of (1). □

Now that we know adding c-defended arguments to a c-admissible extension causes no harm, it is only a small step to a semantics where the extensions include all arguments they defend.

Definition 2.24 (c-complete and c-grounded)**.** The Dung-style complete semantics is defined for any AF $F = (A, R)$ as the set

$$co^c(F) \coloneqq \{E \in ad^c(F) \mid E = \Gamma(E)\}$$

of c-admissible fixpoints of the classic Defense-Operator Γ and the set of all $\subseteq$-minimal fixpoints of Γ is

$$gr^c(F) \coloneqq \{E \in co^c(F) \mid \forall D \in co^c(F) : D \subseteq E \Rightarrow E = D\}$$

the Dung-style grounded semantics.

Applying these semantics to our example has the following results.

Example 2.25. Example 2.22 continued. Because a c-defends d, $\{a\}$ is not c-complete. The c-complete extensions are $\{a, d\}$, $\{b, d\}$ and the empty set, of which the empty set is c-grounded.

Note that the empty set is not always c-complete, for a non-trivial c-grounded extension consider Example 2.2, where a is unattacked and thus $\{a\}$ is c-grounded, while $\emptyset$ is c-admissible but not c-complete. One could equivalently define c-complete extensions to be conflictfree fixpoints of Γ as fixpoints of Γ always c-defend themselves. From this point of view their existence seems no longer trivial. The following proposition is a consequence of the Fundamental Lemma.

Proposition 2.26. *Let $F = (A, R)$ be an AF. (I) For any c-admissible extension $E \in ad^c(F)$ exists an $E' \in co^c(F)$ such that $E \subseteq E'$.*
(II) The c-grounded extension always exists, is unique and is $G = \bigcap_{E \in co^c(F)} E$.

Proof. By iterating Γ over E (I) follows from the Fundamental Lemma because A is finite.
(II) follows from the fact that the empty set is c-admissible and a subset of any c-complete extension. Thus the fixpoint G resulting from iterating Γ over $\emptyset$ is c-complete by the Fundamental Lemma and a subset of any other fixpoint by Prop.2.20. □

The existence and uniqueness of the grounded extension can also be concluded from applying the Knaster-Tarski Fixpoint-Theorem to the monotonic Γ-Operator on the poset $(2^A, \subseteq)$.[8] On the other hand there is no unique maximal c-complete extension in general, because the maximal fixpoint of Γ on an AF F is usually not conflictfree(for Example 2.2 it is $\{a, c, d\}$). The maximal c-complete extensions coincide with the maximal c-admissible extensions and constitute yet another classic semantics, the c-preferred semantics.

Definition 2.27. (c-preferred) Let $F = (A, R)$ be an AF and $E \subseteq A$. E is a Dung-style preferred extension of F, $E \in pref^c(F)$ iff $E \in ad^c(F)$ and E is maximal w.r.t. set inclusion in $ad^c(F)$ that is for any $D \in ad^c(F)$ it holds that if $E \subseteq D$ then $E = D$.

Example 2.28. $\{a, d\}$ and $\{b, d\}$ are the two c-preferred extensions of Example 2.22.

Proposition 2.29. *Let $F = (A, R)$ be an AF. Then the c-preferred extensions of F are exactly the maximal c-complete extensions of F which means* $pref^c(F) = \{E \in co^c(F) \mid \forall D \in co^c(F) : E \subseteq D \Rightarrow E = D\}$.

Proof. Follows directly from Prop.2.26(I). □

The last of Dungs five semantics was designed with the stable models of logic programming in mind. The idea is to decide for every! argument whether it is accepted or rejected. A c-stable extension thus has to attack every argument it does not contain. In terms of labelings this amounts to having only the two values *in* or *out*, like in classic binary logic.

Definition 2.30 (c-stable)**.** Let $F = (A, R)$ be an AF. The set of all Dung-style stable extensions of F is

$$stb^c(F) = \{E \in ad^c(F) \mid E \cup E^+ = A\}$$

the set of all c-admissible extensions which attack any argument they do not contain.

Example 2.31. Example 2.22 continued. $\{a, d\}$ is a c-stable extension, because a attacks both remaining arguments b, c.

[8] [Tarski, 1955], see also Section 4.2

C-stable extensions are always c-preferred,[9] while the reverse is not true. For example $\{a\}$ is a c-preferred extension of Example 2.2 but neither attacks c nor d. The c-stable semantics is also missing a useful property the other c-semantics have - directionality.[10]

Definition 2.32. An extension-based semantics ς satisfies *directionality* iff

$$\varsigma(F\downarrow_U) = U \cap \varsigma(F)$$

for any $F = (A, R) \in U_F$ and any unattacked argument set $U \subseteq A$, $U^- \subseteq U$.[11]

Proposition 2.33. *ad^c, co^c, gr^c and $pref^c$ satisfy directionality, stb^c does not.*[12]

2.3 Various weak semantics and their respective underlying concepts

The classic concept of defense by attack grasps very well the dialogical form argumentation between humans often takes, but sometimes it is considered too strict. For instance, some researchers argue c should be acceptable in Example 2.2 and others that it should also be acceptable in Example 2.22. Semantics which solve such problems are commonly referred to as weak semantics because in order to do so they have to accept more arguments than c-semantics so their criteria for accepting arguments have to be "weaker" than c-admissibility. It is important to note that THE weak argumentation semantics does not exist. The various weak semantics in existence were developed with different intentions and focus on different problematic aspects of c-semantics so the name weak semantics has become misleading. The term "weakly complete" for example is by now used for two different semantics, the weakly complete semantics of [Dondio and Longo, 2021] and the weakly complete semantics of [Baumann et al., 2020b]. In order to avoid confusion we refrain from using "weak" in the name of any semantics in this work and decided to name them after the principles motivating them instead. The great structural differences in the design of weak semantics and their sheer number made it impossible to integrate all weak semantics from the works we mentioned in the introduction, so we limited ourselves to three different semantics families for a comparative study under the new defense notion, to which a fourth, new one, is added in Chapter 3.

---

[9] for proof see [Dung, 1995]

[10] This property becomes relevant in the proofs of Section 5.2 but as it is also satisfied by the majority of the semantics in this work, we decided to introduce it early on

[11] [Baroni and Giacomin, 2007]

[12] [Baroni and Giacomin, 2007]

2.3.1 The relative defense of Bodanza and Tohmé

The first weak semantics we would like to introduce here is the cogent semantics proposed by [Bodanza and Tohmé, 2009]. Motivated by dialogue games, they took a very unique approach on the issue of self-attackers and interpreted sets of arguments as argumentation strategies. The idea behind this is to compare extensions directly with each other and determine admissibility relative to a potential opposing extension. This is done by reducing a given AF to the two extensions in question and checking them for classic admissibility in the reduced AF.

Definition 2.34 (cogent semantics)**.** Let $F = (A, R)$ be an AF and $E, D \subseteq A$ argument sets. E is *at least as cogent as* D iff E is c-admissible in $F{\downarrow}_{E \cup D}$. An $E \subseteq A$ is a *cogent* extension, $E \in cog(F)$ iff E is at least as cogent as D for every $D \subseteq A$ that is at least as cogent as E.
E is a *sustainable* extension iff E is maximal w.r.t. set inclusion in $cog(F)$.

Basically admissibility is weakened at the extension-level while classic admissibility is still applied on argument level during the process. As a result these two semantics, cogent and its preferred form sustainable semantics, are much closer to c-semantics than other weak semantics and show a very similar behavior. We will go in more detail about this in the following chapters, for now we only demonstrate how cogency impacts admissibility with a simple example.

Example 2.35. The set of all cogent extensions for Example 2.2 is
$cog(F) = \{\emptyset; \{a\}; \{a, c\}; \{c\}\}$, of which $\{a, c\}$ is the only sustainable extension.
$\{a\}$ and the empty set are cogent because they are c-admissible.
$\{c\}$ is a cogent extensions of F because $\{d\}$ is not c-admissible in $F{\downarrow}_{\{c,d\}}$ so $\{d\}$ is not at least as cogent as $\{c\}$. In $F{\downarrow}_{\{b,c\}}$ both singletons are c-admissible, so $\{c\}$ is at least as cogent as $\{b\}$. The same goes for other argument sets, unless d is contained, $\{c\}$ is always at least as cogent as the other set, and if d is contained, the other set cannot be at least as cogent as $\{c\}$. This also holds true for $\{a, c\}$.
$\{b\}$ on the other hand is not cogent, as $\{a\}$ is c-admissible in $F{\downarrow}_{\{a,b\}}$ and $\{b\}$ is not. $\{d\}$ is not cogent because the empty set is c-admissible in $F{\downarrow}_{\{d\}}$ and $\{d\}$ is not.

Two other weak semantics are given in [Bodanza et al., 2014], the so called lax and tolerant semantics.[13] We will not include these two in the following chapters due to the reasons already mentioned. Nevertheless, we decided to introduce tolerant semantics at least, since it solves an aspect of weak argumentation other

[13] Tolerant semantics were already introduced in the original paper [Bodanza and Tohmé, 2009]

approaches do not investigate. While lax semantics is simply a further weakened form of cogent semantics, the tolerant semantics is based on the peculiar concept of cyclic cogency.

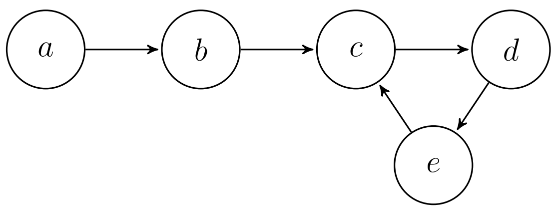
Definition 2.36. Let $F = (A, R)$ be an AF and $E \subseteq A$. E is *cyclically cogent* iff every $D \subseteq A$ satisfies at least one of the following conditions:

1. D is not at least as cogent as E
2. E is at least as cogent as D
3. A chain of arguments sets $D_1 = D, D_2, ..., D_n = E,\ D_i \subseteq A$ exists such that D_{i+1} is at least as cogent as D_i but D_i not at least as cogent as D_{i+1} for all $i \in (1, ..., n)$.[14]

E is *tolerant* iff E is a maximal cyclically cogent extension w.r.t. set inclusion.

The idea is for an extension to defend itself over a chain of extensions that each defeat their predecessor, in the sense that the predecessor is not c-admissible in the resp. restriction. This is an attempt to solve the problem of non-admissibility in odd cycles. A cycle is a closed chain of arguments attacking each other. A self-attacker, for example, is a cycle of length one. The philosophy behind cyclic-cogency is that arguments in cycles should be treated the same no matter whether the cycle is odd or even. C-semantics, however, only accepts arguments in even cycles (like a in Example 2.22). Cyclic-cogency tries to fix this, with mixed results, as the following example shows.

Example 2.37. We explain only one example of a cyclically cogent extension for this AF, the extension $\{c\}$.



$\{c\}$ is at least as cogent as any of the sets $\emptyset, \{a\}$, $\{d\}$, $\{a, d\}$, $\{a, c\}$. For the sets $\{b\}$, $\{e\}$, $\{b, d\}$, $\{b, e\}$ and $\{a, e\}$ we need the cyclic cogency criterion. The rest is not conflictfree and therefore not at least as cogent as $\{c\}$.
For $\{e\}$ the chain $(\{c\}, \{d\}, \{e\})$ satisfies Def. 2.36, this also works for $\{a, e\}$. For $\{b\}$ use $(\{c\}, \{a, d\}, \{b\})$. For $\{b, d\}$ it gets slightly more complicated with $(\{c\}, \{d\}, \{e\}, \{a, c\}, \{b, d\})$ and for $(\{b, e\})$ the chain $(\{c\}, \{a, d\}, \{b, e\})$ does the trick. So $\{c\}$ is cyclically cogent.

[14]The tuple $(1, ..., n)$ is used to denote the set $\{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ for a natural number $n \in \mathbb{N}$

While c can defend itself against e with this approach as planned, it can also defend itself against b despite b not being in a cycle with c. On top of that c could not defend itself against b if it did not attack *anything*. Even so (and exactly because) this semantics succeeds in solving the original problem, the side effects of this unique defense need some further investigation.

2.3.2 The undecidedness blocking of Dondio and Longo

Motivated by their research in ambiguity blocking P.Dondio and L.Longo followed a different approach to weak semantics. They focused on the behavior of the undecided-status in labelings. Their reasoning is that undecidedness should not be hereditary as it is the case with Dung-style semantics. To counter the propagation of the undecided-label they developed a set of liberal labeling rules leaving open the possibility to accept arguments only attacked by undecided (and out) labeled arguments or not. The result was the following family of *liberal undecidedness-blocking(lub)*-semantics from [Dondio and Longo, 2021]. [15]

Definition 2.38. (lub-labelings) Let $F = (A, R)$ be an AF. $lab : A \rightarrow \{in, out, undec\}$ is a *lub-complete* labeling on F iff for every argument $a \in A$:

(1) If $lab(a) = in$ then $lab(b) \neq in$ for every attacker $b \in \{a\}^-$

(2) If $lab(a) = out$ then $lab(b) = in$ for at least one attacker $b \in \{a\}^-$

(3) If $lab(a) = undec$ then $lab(b) \neq in$ for every attacker $b \in \{a\}^-$ and for at least one attacker $lab(b) = undec$

lab is *lub-preferred* iff $lab \in co^{lub}(F)$ and the set of all in-labeled arguments $in(lab)$ is maximal w.r.t. $\subseteq$ among all lub-complete labelings on F.

In addition to these two semantics the c-grounded and c-stable semantics can be used as the respective lub-grounded and lub-stable semantics, which has been shown in [Dondio and Longo, 2021]. Lub-semantics do not only accept extensions with self-attacking attackers but also those with conflicting attackers, like the attackers a and b of c from Example 2.22. Let us explain this in detail.

[15] As the name ub-complete is taken by the SCC-recursive variant and the name weakly complete is misleading, I will refer to the weakly complete semantics from [Dondio and Longo, 2021] as the lub-semantics.

Example 2.39. Example 2.22 continued. Apart from the c-complete labelings,[16] the labelings lab_1 with $lab_1(c) = in, lab_1(d) = out, lab_1(a) = lab_1(b) = undec$ and lab_2 with $lab_2(d) = in$ and $lab_2(a) = lab_2(b) = lab_2(c) = undec$ are lub-complete because the in-labeled argument c (d resp.) is only attacked by undecided arguments. a and b can only be labeled one in and one out or both undecided according to rule (3) of Def. 2.38. In the latter case their undecided-status can then be blocked at c (lab_1) or propagated to it (lab_2).
lab_1 is also lub-preferred. The other lub-preferred labelings are those with $in(lab) = \{a, d\}$ and $in(lab) = \{b, d\}$ resp..

One can say that the lub-semantics succeed in maintaining most of the features of Dung-style-semantics, like reinstatement and directionality, while forsaking their core - the principle of admissibility. This becomes evident in the missing lub-admissible semantics. It leaves open the question what concept of defense replaces admissibility in lub-semantics. In order to investigate this, we conduct a translation to extension-based semantics here. This task is simplified thanks to lub-complete semantics satisfying the rejection property.[17]

Proposition 2.40 (lub-complete extensions)**.** *Let $F = (A, R)$ be an AF, $E \subseteq A$. For E exists an lub-complete labeling $lab \in co^{lub}(F)$ such that $E = in(lab)$ iff E is conflictfree and $\Gamma(E) \subseteq E$.*

To put it simply, an extension of this semantics does not have to defend itself against attackers, but it has to contain any argument that it does actually defend. For example we have $\Gamma(\{c\}) = \emptyset \subseteq \{c\}$ in Example 2.22 for which lab_1 from Example 2.39 with $in(lab_1) = \{c\}$ is an lub-complete labeling. It follows the proof for the general case.

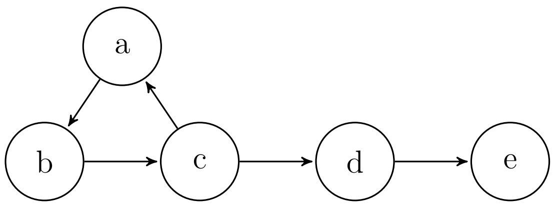
Proof. ($\Rightarrow$) Let lab be an lub-complete labeling for $F = (A, R)$. Then $in(lab)$ is conflictfree because of (1) in Def. 2.38 and for any $a \in A$ with $lab(b) = out$ for all $b \in \{a\}^-$ (3) implies that $lab(a) \neq undec$ and (2) that $lab(a) \neq out$, therefore $lab(a) = in$ for any $a \in \Gamma(in(lab))$.
($\Leftarrow$) Let E be a conflictfree extension of $F = (A, R)$ with $\Gamma(E) \subseteq E$. Then the labeling lab defined by $in(lab) = E$, $out(lab) = E^+$ and $undec(lab) = A \setminus (E \cup E^+)$ satisfies condition 1, because E is conflictfree. It satisfies condition 2, because only arguments attacked by E are labeled *out*. And it satisfies condition 3, as all arguments attacked by E are labeled *out*, not *undec*, and all arguments only attacked by *out* arguments are elements of $\Gamma(E) \subseteq E$, therefore *in*, not *undec*. □

[16] All labelings lab where $in(lab) = E$ is a c-complete extension(see Example 2.25)

[17] The rejection property for lub-labelings is a result from [Dondio and Longo, 2021], it follows directly from the labeling-rules in Def. 2.38

Example 2.22 is also an excellent demonstration why the lub-preferred semantics does not satisfy directionality while the lub-complete does.[18] The intersection of the lub-preferred extension $\{c\}$ with the unattacked set $U = \{a, b\}$ is empty, but the emptyset is not an lub-preferred extension of U. We chose the name *liberal* ub-semantics based on another, related property of lub-semantics the developers are aware of themselves and which they demonstrate in [Dondio and Longo, 2021] by the following example. It has two lub-admissible extensions, $\{d\}$ and $\{e\}$.

Example 2.41.



In a situation like this d appears to have a much stronger reason to be accepted than e because d is only attacked by arguments locked in a contradiction which therefore are not part of any lub-complete extension. In contrast e is attacked by said d that *is* lub-complete. The liberal rule of blocking the propagation of the undecided-label at any point of a chain of undecided arguments leads to an unintuitive result here. This concern would become even more obvious if we had a self-attacker attacking d instead of an odd cycle. Accepting an argument d only attacked by a self-contradicting argument, like cogent semantics does, makes sense, but accepting arguments attacked by d, too, seems inconsistent.

The authors of [Dondio and Longo, 2021] propose an elaborate solution for this problem, the SCC-recursive ub-grounded,-complete and -preferred semantics. One of their goals is to block undecidedness "as early as possible" and in order to do this they use the natural order of SCCs in an AF. Introducing SCC-theory would require a way longer introduction than we have already given here, so in order to focus on our main objective, the generalization of weak defense concepts and their comparison, we will not discuss their solution in detail here. Instead we want to propose a different solution for Example 2.41 in the form of a reduct-based semantics, which we define in Section 3.5.

[18] For a proof of this statement see [Dondio and Longo, 2021]

2.3.3 The recursive semantics of Baumann, Brewka and Ulbricht

In [Baumann et al., 2020b] the authors formalize what I would describe as a concept of *no admissible attacker (naa)*. The idea is to have an extension defend itself only against attackers, that could be part of an acceptable extension themselves. The problem of self-attackers is tackled directly by this, since self-attackers and any extension containing them are not conflictfree. As plausible as this approach may sound, its formalization requires some groundwork. The first step is to identify all attackers which classic defense cannot handle.

Definition 2.42 (reduct)**.** Let $F = (A, R)$ be an AF and $E \subseteq A$. The *reduct* of F w.r.t. E is the AF $F^E := \{a \in A \mid a \notin (E \cup E^+)\}$ with the respective restricted attack relation.

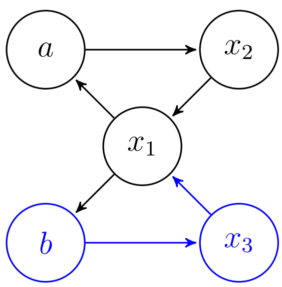
The reduct is best understood as the set of arguments independent from E, the arguments out of its range.[19] If E is attacked by an argument from the reduct it cannot classically defend itself against it. In some but not all cases we want to accept E despite this, so we now need a criterion stating which of the attackers from the reduct we can ignore and which not.

Definition 2.43 (naa-admissible)**.** Let $F = (A, R)$ be an AF and $E \subseteq A$. E is an *naa-admissible* extension, $E \in ad^{naa}(F)$ iff E is conflictfree and for every attacker $b \in E^-$ one of the following holds:

- b is attacked by E, that is $b \in E^+$
- b is not naa-admissible in F^E, that is $b \notin_{ext} ad^{naa}(F^E)$.

In order to develop an understanding how naa-admissibility discriminates between dangerous and harmless attackers the following example from [Dauphin et al., 2021] will prove useful. Its only c-admissible extension is the empty set.

Example 2.44. The reduct of the following AF w.r.t. $\{b\}$ is $F^{\{b\}} = \{a, x_1, x_2\}$ the upper 3-cycle of arguments in black, while the range of $\{b\}$ (in blue) consisting of b itself and $\{b\}^+ = \{x_3\}$ is deleted.



[19] in both the literal and the formal meaning of the word

The argument set $\{b\}$ is naa-admissible because its only attacker x_1 is not naa-admissible in $F^{\{b\}}$. To see this, the next reduct $F^{\{b\}^{\{x_1\}}} = \{x_2\}$ has to be considered. Its only argument x_2 is naa-admissible in $F^{\{b\}^{\{x_1\}}}$ as it is unattacked. It attacks x_1 in $F^{\{b\}}$, so $\{x_1\}$ is not naa-admissible in $F^{\{b\}}$. Any other argument set containing x_1 in $F^{\{b\}}$ is not conflictfree, therefore $x_1 \notin_{ext} ad^{naa}(F^{\{b\}})$, so $\{b\} \in ad^{naa}(F)$.
With the same argumentation $\{a\}$ is naa-admissible, while $\{a, b\}$ is not, because x_1 is unattacked and therefore naa-admissible in $F^{\{a,b\}} = \{x_1\}$. The total set of naa-admissible extensions for this AF is $ad^{naa}(F) = \{\emptyset; \{a\}; \{b\}\}$.

Naa-admissibility is build around the property of odd cycles to have no c-admissible extensions (because there is always this one remaining attacker). It works especially good against self-attackers rendering them completely irrelevant for the acceptability of other arguments.[20] Another group of attackers that pose no threat under naa-admisibility are those attacked by naa-admissible supersets of an extension. In Example 2.22 $\{d\}$ is an naa-admissible extension because c cannot defend itself against a (or b) in the reduct $F^{\{d\}}$. Like for any of the other semantics introduced so far, the $\subseteq$-maximal naa-admissible extensions form their own semantics, the naa-preferred semantics. For Example 2.44 the extensions $\{a\}$ and $\{b\}$ are naa-preferred.

Definition 2.45 (naa-preferred)**.** Let $F = (A, R)$ be an AF. The set of *naa-preferred* extensions of F is

$$pref^{naa}(F) := \{E \in ad^{naa} \mid \forall D \in ad^{naa}(F) : E \subseteq D \Rightarrow E = D\}$$

the set of all $\subseteq$-maximal naa-admissible extensions.

The difference between the naa-semantics from [Baumann et al., 2020b] and the other weak semantics is not so much its recursive schema (in a sense SCC-recursive semantics are recursive too), but that a proper concept of weak defense for naa-semantics is formulated in [Baumann et al., 2020b].

Definition 2.46 (naa-defense)**.** Let $F = (A, R)$ be an AF. An argument set $E \subseteq A$ *naa-defends* another set $X \subseteq A$ iff for every $y \in X^-$ attacker of X one of the following holds:

1. $E \to y$

[20]Because self-attackers are not conflictfree, they are never naa-admissible and thus also never impact the admissibility of any attacker in any reduct

2. $y \notin E$, $y \notin_{ext} ad^{naa}(F^E)$ and some superset $X' \supseteq X$ exists which is naa-admissible in the original AF ($X' \in ad^{naa}(F)$).

A complication of computing admissibility recursively with the reduct is that defended arguments can no longer be simply added to an extension like it was the case with c-defense. The following example serves to illustrate this problem. Therefore naa-defense was defined on set-level instead of argument-level and defense option (2) of Def. 2.46 explicitly requests that naa-admissibility is satisfied (in the original AF!) by the defended set to begin with.

Example 2.47. Example 2.44 continued. The empty set naa-defends $\{a\}$ and $\{b\}$ as both are (subsets of) naa-admissible extensions and their only attacker x_1 is not naa-admissible in $F^\emptyset = F$, but it does not defend $\{a, b\}$, because no naa-admissible extension containing both a and b exists.

As demonstrated the Fundamental Lemma is not valid for naa-defense, so naa-completeness had to be described differently. Instead of containing all arguments or, to be correct, all argument sets it defends under naa-defense, an naa-complete extension E only has to contain those sets it is compatible with, in simpler terms, the naa-defended sets of which E is a subset.

Definition 2.48 (naa-complete and naa-grounded)**.** Let $F = (A, R)$ be an AF. An $E \subseteq A$ is *naa-complete* iff $E \in ad^{naa}(F)$ and E contains every superset $X \supseteq E$ it naa-defends.
It is *naa-grounded* iff $E \in co^{naa}(F)$ and additionally no proper naa-complete subset $D \subset E$ exists that is if E is $\subseteq$-minimal in $co^{naa}(F)$.

For our running example that means:

Example 2.49. Example 2.44 continued. Because the empty set naa-defends a superset of itself, e.g. $\{a\}$, it is not naa-complete.
Since no naa-admissible supersets of them exist,[21] the remaining naa-admissible extensions $\{a\}$ and $\{b\}$ are both naa-complete and, since the empty set is not naa-complete, naa-grounded.

As the example proves, there is no unique naa-grounded extension in general. From the definition of naa-defense and naa-completeness it is not clear at all whether naa-complete extensions exist for every AF. This question can be answered positively thanks to the following important result.

[21]Option 1 of Def. 2.46 does not require an naa-admissible superset but there are also no supersets of $\{a\}$ that can be defended solely with this option, for example $\{a, x_1\}$ is attacked by x_3 which is not attacked by a

Proposition 2.50. *Let $F = (A, R)$ be an AF. Then*

$$pref^{naa}(F) = \{E \in co^{naa}(F) \mid \forall D \in co^{naa}(F) : E \subseteq D \Rightarrow E = D\}$$

the naa-preferred extensions of F coincide with the $\subseteq$-maximal naa-complete extensions of F.[22]

Instead of providing the already known proof for this we want to use the remaining space for some notable results related to naa-semantics. First, ad^{naa} and $pref^{naa}$ satisfy directionality, while co^{naa} and gr^{naa} do not [Baumann et al., 2020a,Dauphin et al., 2020]. Next, we want to introduce the property that replaces the Fundamental Lemma for naa-semantics, namely modularization, from [Baumann et al., 2020a].

Definition 2.51 (modularization)**.** A semantics ς satisfies *modularization* iff for any AF $F = (A, R)$ and any $E, D \subseteq A$ the following holds: If $E \in \varsigma(F)$ and $D \in \varsigma(F^E)$ then $E \cup D \in \varsigma(F)$

Modularization explains under which circumstances an argument a *can* be added to an naa-admissible extension E without loosing naa-admissibility. It has to be naa-admissible in the reduct and one of the naa-extensions of the reduct D containing it has to be added as a whole to E. This is always possible in naa-semantics according to the following proposition that has already been proven in [Baumann et al., 2020a].

Proposition 2.52. *ad^{naa}, co^{naa}, gr^{naa} and $pref^{naa}$ satisfy modularization.*

Modularization itself is already a powerful property but it also has the following useful consequence.

Proposition 2.53 (persisting non-admissibility)**.** *Let $F = (A, R)$ be an AF, ς a semantics satisfying modularization and $a \notin_{ext} \varsigma(F)$. Then $a \notin_{ext} \varsigma(F^E)$ for any $E \in \varsigma(AF)$.*

An argument that is not naa-admissible in the main AF is also not naa-admissible in the reduct of any naa-admissible extension of this AF. This persisting non-admissibility will become a great help in some proofs later in this work.

Proof. By contradiction. Suppose some $D \in \varsigma(F^E)$ existed with $a \in D$, then $E \cup D \in \varsigma(F)$ because of modularization, so a would be in a ς-extension of F to begin with. $\square$

[22] for a proof see Theorem 5.3 of [Baumann et al., 2020b]

The works of Dauphin et al. have successively contributed to the understanding of naa-semantics and introduce a number of other weak semantics inspired by it: the qualified and semiqualified semantics found in [Dauphin et al., 2020] as well as the three alternative weak defense forms build on naa-admissibility and their corresponding semantics families in [Dauphin et al., 2021]. Although we do not include any of these semantics in the later chapters for reasons already stated, we want to use the opportunity to give an answer to one of the open questions from [Dauphin et al., 2021].

Naa-defense does not request naa-admissibility from the defended set, it only has to be contained in an naa-admissible extension. This hinders an naa-admissible extension from being naa-complete in some cases where certain arguments belonging to an extension are defended but not all, so completeness is only reached by adding more than one does defend. The ∃-defense was designed to deny such problematic sets the status of being defended.

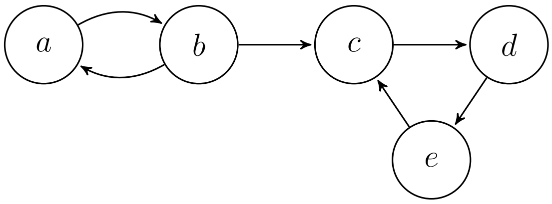
Definition 2.54 (∃-defense)**.** Let $F = (A, R)$ be an AF. An argument set $E \subseteq A$ ∃-*defends* another set $X \subseteq A$ iff for every $y \in X^-$ attacker of X one of the following holds:

1. $E \rightarrow y$
2. $y \notin_{ext} ad^{naa}(F^E)$ and $E \cup X \in ad^{naa}(F)$

E is ∃-complete iff $E \in ad^{naa}(F)$ and E contains every superset $X \supseteq E$ it ∃-defends and it is ∃-grounded iff it is ⊆-minimal in $co^{\exists}(F)$.[23]

Where ∃-defense makes a difference, shows the following example from [Dauphin et al., 2021].

Example 2.55. In the AF below $\{d\}$ is naa-defended but not ∃-defended by the empty set.



It was conjectured in [Dauphin et al., 2021] that both the ∃-complete and the ∃-grounded semantics satisfy directionality. We will now prove this is only true for $co^{\exists}$.

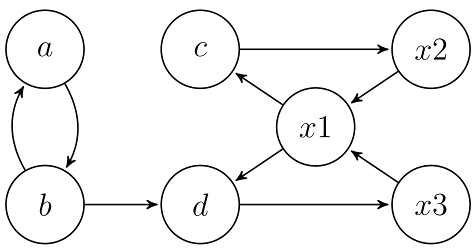
[23] [Dauphin et al., 2021]

Theorem 2.56. *The $\exists$-complete semantics satisfies directionality, the $\exists$-grounded does not.*

Proof. Let $F = (A, R)$ be an AF, $U \subseteq A$ an unattacked set and $E \subseteq A$.
Suppose first E is $\exists$-complete on F, but $E \cap U$ is not on U. Because the naa-admissible semantics satisfies directionality by Prop. 4.16 of [Baumann et al., 2020a], $E \cap U$ is naa-admissible. So if $E \cap U$ is not $\exists$-complete, then some $X \supset E \cup U$ in U exists, which is naa-admissible in U and naa-defended by $E \cup U$ in U. By Prop. 4.11 of [Baumann et al., 2020a] $X \setminus E$ is naa-admissible in $U^{E \cap U}$ and because of Prop. 4.16 $X \setminus E$ is also naa-admissible in F^E. Because of modularization (Cor. 4.2) $E \cup X \setminus E$ is therefore an naa-admissible superset of E in F and since all attackers y of X are in U and $E \cap U$ naa-defends X in U, we have by directionality $y \notin_{ext} ad^{naa}(F^E) \Leftrightarrow y \notin_{ext} ad^{naa}(U^{E \cap U})$ so E naa-defends $E \cup X$ which is a proper superset of E that is naa-admissible. Therefore E is not $\exists$-complete in F. Contradiction.
Suppose now $E \subseteq U$ is $\exists$-complete on U. Then either E is $\exists$-complete on F too, or some superset $X \supset E$ in F exists, which is naa-admissible and naa-defended by E. We know $(X \setminus E) \cap U = \emptyset$, because otherwise E would not be $\exists$-complete in U (directionality with regard to $X \cap U$. Therefore X satisfies $X \cup U = E$ and is naa-admissible in F. If X is $\exists$-complete in F we are done, if not we can repeat this argument for the then existing superset $X' \supset X$ which is by the conditions of $\exists$-completeness again naa-admissible and satisfies $X' \cup U = E$. Since we only consider finite AFs this extending procedure terminates at an $\exists$-complete extension of F. This concludes the proof of directionality for the $\exists$-complete semantics.

□

Counterexample 2.57. Let $F = (A, R)$ be the AF below and $U = \{a, b\}$ the unattacked set, then $gr^{\exists}(U) = \{\emptyset\}$.
But $\{a, d\}$ is $\exists$-grounded in F and $U \cap \{a, d\} \neq \emptyset$.



Chapter 3

Generalizing defense

From an early point on we started looking for a common ground of weak semantics, a basic concept they can be traced back to. Instead of asking ourselves what the ideal weak semantics looks like, we wanted something that describes the individual "weakness" of any semantics in a way that allows an impartial comparison and a categorization of different approaches. The strongly varying methods of the semantics in Section 2.3 should give the reader an idea of the dimension such a project has. So what is "weakness" in terms of abstract argumentation? Weakness can concern a variety of attributes, but when it comes to weak semantics it means the *defense* of the accepted arguments is weak. And what is defense, generally speaking? Our answer to this question is that defense is about deflecting rsp. rejecting rsp. neutralizing incoming attacks. Therefore the defense notion of a semantics should be able to exactly specify which attacks can be rejected and which not. The formalization of this idea leads us back to the roots - the classic defense by Dung.

3.1 Rethinking classic defense

The more time we spent on other defense concepts - like the three introduced in Section 2.3 - the higher we valued Dungs classic defense. Not only is *defense by attack* an intuitive principle, it was also formalized as an operator which makes it possible to analyze his semantics in terms of algebra and lattice theory. Another important feature of defense in the context of abstract argumentation is completeness. The introduction of c-complete extensions as fixpoints is the most elegant way of defining completeness we have come across so far. It is also invaluable for embedding classic semantics into related research like logic programming where fixpoint theory is held in high regard.

If the Γ-Operator has one flaw it is defending too much. A self-attacker defends itself against itself with this operator but this kind of defense cannot be considered successful in argumentation. In case of a successful defense no threat for the defended arguments should be left. So instead of the Γ-Operator consider the following operator for classic defense.

Definition 3.1. (refined c-defense) Let $F = (A, R)$ be an AF. We define an alternative classic defense operator $\chi_c : 2^A \to 2^A$ with

$$\chi_c(E) := \{a \in A \setminus E^+ \mid a \textit{ unattacked in } F{\downarrow}_{A \setminus E^+}\}$$

for any $E \subseteq A$.

The difference between the two operators lies in excluding arguments attacked by an extension from being defended by it, too. The following example serves to illustrate this point.

Example 3.2. In Example 2.2 the singleton $\{d\}$ c-defends itself by the classic notion as d attacks its only attacker, d, itself. Therefore $\Gamma(\{d\}) = \{a, d\}$. For the refined c-defense on the other hand d can only defend arguments in $A \setminus \{d\}^+$, so $d \notin \chi_c(\{d\})$. The arguments d does not attack are a and b of which only a is unattacked, thus $\chi_c(\{d\}) = \{a\}$.

With this operator conflictfreeness is no longer a separate criterion. While the Γ-Operator allows for sets with conflicts to defend themselves and only satisfies the Fundamental Lemma, the new operator χ_c satisfies $E \subseteq \chi_c(E)$ only for conflictfree sets of arguments in the first place.

Proposition 3.3. *Let $F = (A, R)$ be an AF. Then for any conflictfree argument set $E \subseteq A$ it holds that $\chi_c(E) = \Gamma(E)$ and any! $E \subseteq A$ is Dung-style-admissible iff $E \subseteq \chi_c(E)$.*

Proof. For the first part supppose $a \in \Gamma(E)$. Then all attackers of a are in turn attacked by E and therefore elements of E^+. Since E is conflictfree, $a \notin E^+$ by the Fundamental Lemma. Therefore $a \in A \setminus E^+$ and all of its attackers not, so a is unattacked in $A \setminus E^+$. The other direction is trivial.
For any set of arguments E it holds that E not conflictfree implies $E \nsubseteq \chi_c(E)$ because in this case some $e \in E$ exists that is not in $A \setminus E^+$ and $\chi_c(E)$ is a subset of the arguments of $A \setminus E^+$. The rest follows from $\Gamma(E) = \chi_c(E)$.

□

Note that the equality of the Γ- and χ_c-Operator for conflictfree sets of arguments includes the identity of their fixpoints (among conflictfree sets). This means the c-complete semantics is compatible with the new defense notion, the same holds for $pref^c$ and gr^c.

Corollary 3.4. *For any AF $F = (A, R)$ we have $co^c(F) = \{E \subseteq A \mid E = \chi_c(E)\}$ and $pref^c$ and gr^c are the maximal rsp. minimal fixpoints of χ_c.*

We now have a defense operator for classic semantics which guarantees the conflictfreeness of defended sets while preserving completeness in form of fixpoints. This was achieved by dividing the defense by attack principle into two steps.

1. Delete all arguments that are attacked by an argument set E (Attack)
2. Determine which of the remaining arguments are now unattacked(Defense)

3.2 Defense by defeat - generalizing the defense operator

Defense is no primary feature of Dungs argumentation formalism. All we have is the attack relation, so defense can only be defined w.r.t. it, how is left open. In order to generalize defense we turn this dependency on its head and say an argument is defended if all its attackers are *defeated*. Now we have a fixed concept of defense and leave open how *defeat* is defined instead. In order to describe e.g. naa-defense in terms of defeat an argument-level relation no longer suffices. Therefore we formalize defeat as an operator on set-level.

Definition 3.5 (defeat operator)**.** Let $U_F^+ := \{(F, E) \mid F = (A, R) \in U_F,\ E \subseteq A\}$ be the set of all pairs of finite AFs $F = (A, R)$ with any of their resp. argument subsets $E \subseteq A$.
A *defeat operator* is a mapping $\delta : U_F^+ \rightarrow 2^{U_{arg}}$ that assigns to each such pair (F, E) an argument set $\delta((F, E)) \subseteq A$ of arguments defeated by E in F. If F is clear, we can simplify δ to the unary operator $\delta : 2^A \rightarrow 2^A$ on F.

We can now apply the defense concept of Section 3.1 to a given defeat operator δ.

1. Delete all arguments $a \in \delta(F, E)$ that are defeated by an argument set E according to δ.(Defeat)
2. Determine which of the remaining arguments are now unattacked(Defense)

The result of this process is a generalized form of defense operators.

Definition 3.6 (defense operator)**.** Let δ be a defeat operator. The *defense operator* $\chi_\delta : 2^A \to 2^A$ induced by δ on an AF $F = (A, R)$ is defined for every $E \subseteq A$ by

$$\chi_\delta(E) := \{a \in Free_\delta \mid a \text{ unattacked in } Free_\delta\}$$

where $Free_\delta := F{\downarrow}_{A\setminus\delta(E)}$ is the restriction of F to all arguments not defeated by E .

Let us demonstrate this concept with the new defense operator for classic semantics we introduced in the previous section.

Proposition 3.7. *For any AF* $F = (A, R)$ χ_c *is the defense operator induced by* $Def_c : 2^A \to 2^A$*,* $Def_c(E) := E^+$ *for any* $E \subseteq A$*.*

Proof. Let $F = (A, R)$ be an AF and $E \subseteq A$. Then the set of all arguments not defeated by E is $Free_c(E) = A \setminus Def_c(E) = A \setminus E^+$. The defense operator induced by Def_c according to Def. 3.6 now assigns to E the set of all arguments unattacked in $Free_c(E)$ which is exactly the set $\{a \in A \setminus E^+ \mid a \text{ unattacked in } F{\downarrow}_{A\setminus E^+}\} = \chi_c(E)$ of arguments defended by E according to Def. 3.1. □

Admissibility can now be generalized as the property of an extension to defend at least itself under a given defense operator.

Definition 3.8 (defeat-based admissibility)**.** Let $F = (A, R)$ be an AF and δ a defeat operator. An argument set $E \subseteq A$ is δ-admissible iff $e \in \chi_\delta(E)$ for every argument $e \in E$ that is E defends at least itself under the defense operator χ_δ induced by δ. Define the δ-admissible semantics as $ad^\delta := \{E \subseteq A \mid E \subseteq \chi_\delta(E)\}$ the set of all δ-admissible extensions of F.

Independently from the defeat operator this concept of admissibility comes with some intrinsic properties.

Proposition 3.9. *Let* $F = (A, R)$ *be an AF and let* δ *be a defeat operator. Then the following holds:*

I. $\emptyset \in ad^\delta(F)$

II. $\chi_\delta(E)$ *is conflictfree for all* $E \subseteq A$

Proof. (I) The empty set is a subset of any other argument set. Even if $\chi_\delta(\emptyset) = \emptyset$ (I) holds true. (II) $\chi_\delta(E)$ is defined as the set of all unattacked arguments in a certain restriction of F. But if all its elements are unattacked there can be no conflicts among them. □

Like under our new defense operator for c-semantics, the admissible extensions derived from a defeat operator are always conflictfree (even if the defeat operator in question does not defeat arguments attacked by an extension). They are also conflictfree with regard to the defeat operator used for generating them, a huge advantage when discussing weak semantics. Because even if an extension is conflictfree there could be indirect conflicts among arguments if a defense concept includes defeating more than one attacks. This cannot happen with our defeat-based defense.

Theorem 3.10 (direct and indirect conflictfreeness)**.** *Let $F = (A, R)$ be an AF and $E \subseteq A$. If there exists a defeat operator δ such that E is δ-admissible ($E \in ad^{\delta}(F)$) then E is conflictfree and $E \cap \delta(E) = \emptyset$.*

Proof. A δ-admissible E satisfies $E \subseteq \chi_\delta(E)$ which is conflictfree by Prop.3.9. For the indirect conflictfreeness suppose some $e \in E \cap \delta(E)$ exists, then $e \notin Free_\delta(E)$ by Def.3.6 and thus $e \notin \chi_\delta(E)$. But then $E \nsubseteq \chi_\delta(E)$. □

We need not stop at admissibility. Any argumentation semantics based on c-admissibility can be generalized to a defeat-based semantics. This includes, most importantly, a generalized complete semantics in form of fixpoints.

Definition 3.11 (defeat-based semantics)**.** An extension-based semantics ς_δ is based on a defeat operator δ iff $\varsigma_\delta(F) \subseteq ad^{\delta}(F)$ for all $F \in U_F$.

Definition 3.12 (generalized semantics family)**.** Let δ be a defeat operator. Define

the δ-complete semantics as the set $co^{\delta}(F) := \{E \subseteq A \mid E = \chi_\delta(E)\}$ of all fixpoints of χ_δ

the δ-grounded semantics $gr^{\delta}(F)$ as the set of all $\subseteq$-minimal E in $co^{\delta}(F)$.

the δ-preferred semantics $pref^{\delta}(F)$ as the set of all $\subseteq$-maximal E in $ad^{\delta}(F)$.

the δ-stable semantics as the set $stb^{\delta}(F) := \{E \in ad^{\delta}(F) \mid E \cup \delta(E) = A\}$ of all extensions which defeat every argument they do not contain

for any $F = (A, R) \in U_F$.

To give a simple and useful example of this besides c-semantics, consider cf-semantics.

Definition 3.13 (conflictfree defeat operator)**.** The conflictfree defeat operator $Def_{cf} : U_F^+ \to 2^{U_{arg}}$ is defined for any AF $F = (A, R) \in U_F$ and any $E \subseteq A$ by

$$Def_{cf}(F, E) \coloneqq A \setminus E$$

and the induced cf-admissible semantics is $ad^{cf}(F) \coloneqq \{E \subseteq A \mid E \subseteq \chi_{cf}(E)\}$

This is the perfect example to see how defeat-based defense realizes conflictfreeness during the second step. The defeat operator of cf-semantics ensures no outside arguments matter by defeating(deleting) them all in the first step(Defeat). The remaining conflicts can only be internal and if they exist, they hinder the extension to be unattacked as whole in the second step(Defense). So an argument set only cf-defends itself if it is conflictfree.

Example 3.14. Example 2.2 continued. For the argument set $E = \{b, c\}$ the set of cf-defeated arguments is $Def_{cf}(E) = \{a, d\}$, so in $Free_{cf}(E) = E$ only c and b remain which are both not unattacked. Therefore $\chi_{cf}(E) = \emptyset$ so E is not cf-admissible.

Proposition 3.15. *Let $F = (A, R)$ be an AF. Then $co^{cf}(F) = ad^{cf}(F) = cf(F)$, $pref^{cf}(F) = na(F)$ and $gr^{cf}(F) = \{\emptyset\}$ for the resp. semantics based on the defeat operator Def_{cf} defined above.*

Proof. This can be shown with reasonable effort by applying Prop. 3.9 and the relevant definitions from Section 2.1. □

In the following sections of this chapter we redefine the three semantics (families) from Section 2.3 as defeat-based semantics by proposing a defeat operator for each of them and demonstrate how a new semantics family can be constructed easily by modifying a defeat operator.

3.3 Naa-complete extensions as natural fixpoints

We start with naa-semantics since the defeat operator we need is basically already given in the definition of naa-admissibility (Def. 2.43).

Definition 3.16 (naa-defeat operator)**.** For every AF $F = (A, R)$ and every set of arguments $E \subseteq A$ in F the naa-defeat operator Def_{naa} is defined as follows: Let $a \in A$. Then $a \in Def_{naa}(F, E)$ iff $a \in E^+$ or $a \in F^E$ and $a \notin_{ext} ad^{naa}(F^E)$.

Naa-admissibility is defined recursively and the same applies to this defeat operator. Only by knowing the naa-admissible extensions of the reduct we are able to compute the set of defeated arguments for a certain extension. Because defeat-based naa-admissibility can be directly derived from the naa-defeat operator and since the reduct is a real subset for any nonempty set of arguments, this description of naa-admissibility is as well-defined as the original naa-admissible semantics. We will now prove that not only the naa-defeat operator describes naa-admissibility correctly but that the naa-complete extensions are exactly the fixpoints of the induced naa-defense operator.

Theorem 3.17. *Let $F = (A, R)$ be an AF and χ_{naa} the defense operator induced by Def_{naa} on F. Then χ_{naa} satisfies for any $E \subseteq A$:*

I. E is naa-admissible iff $E \subseteq \chi_{naa}(E)$

II. E is naa-admissible $\Rightarrow \chi_{naa}(E) = \bigcup\{X \supseteq E \mid E\, naa - defends\, X\}$

III. E is naa-complete iff $E = \chi_{naa}(E)$

Proof. (I) Because of $E^+ \subseteq Def_{naa}(E)$ it follows again that E has to be conflictfree for $E \subseteq \chi_{naa}(E)$ to hold. If E is naa-admissible then by Def. 2.43 and 3.16 all its attackers are elements of $Def_{naa}(E)$ so E is unattacked in $Free_{naa}(E)$ and vice versa.
(II) ($\subseteq$) Suppose $a \in \chi_{naa}(E)$ then $a \in Free_{naa}(E)$ so a is not attacked by E and there exists an extension $D \in ad^{naa}(F^E)$ with $a \in D$. By modularization (Def. 2.51) we have $E \cup \{a\} \subseteq E \cup D \in ad^{naa}(F)$, so an naa-admissible superset of $E \cup \{a\}$ exists. Now since a is unattacked in $Free_{naa}(E)$ for every attacker y of a it holds $y \in Def_{naa}(E) = E^+ \cup \{y \in F^E \mid y \notin ad^{naa}(F^E)\}$ so either condition 2) or 1) of naa-defense is satisfied for every attacker of $\{a\}$ and since E is naa-admissible the same holds for E. Therefore for every element a of $\chi_{naa}(E)$ $E \cup \{a\}$ is an naa-defended superset of E.
($\supseteq$) Suppose X is an naa-defended superset of E, then by proposition 4.11 of

[Baumann et al., 2020a] the set $X \setminus E$ has an naa-admissible superset in F^E, so $X \subseteq Free_{naa}(E)$. Since X is naa-defended by E all of its attackers are defeated by E. Therefore X is unattacked in $Free_{naa}(E)$, so $X \subseteq \chi_{naa}(E)$.
(III) Suppose E is naa-complete. Then E has no proper superset that is naa-defended by E, so $E = \bigcup\{X \supseteq E \mid E\ naa-defends\ X\}$ and since E has to be naa-admissible in order to be naa-complete we can apply (II) so $E = \chi_{naa}(E)$. For the other direction we can again apply (II) because if $E = \chi_{naa}(E)$, E is naa-admissible according to (I). □

Both the fact that naa-defense had to be defined on set-level and the fact that the naa-complete and -grounded semantics do not satisfy directionality made naa-defense look like a different matter from naa-admissibility. But it turns out naa-complete extensions are the natural fixpoints of a defeat operator directly derived from the definition of naa-admissibility. We have proved that the naa-semantics family was well-defined as a whole to begin with. In this new light Prop. 2.50, the coincidence of maximal naa-complete and naa-preferred extensions, becomes a surprising result. This coincidence was thought to be a design choice embedded in defense condition (2) of Def. 2.46. Now we know that naa-defense has nothing to do with this, because the naa-defeat operator is based on naa-admissibility only. The discussion of this topic is continued in Section 4.3. We conclude this section with the introduction of the naa-stable semantics which, as to be expected because of modularization, coincides with the naa-preferred semantics.

Proposition 3.18. *Let $F = (A, R)$ be an AF. Then $stb^{naa}(F) = pref^{naa}(F)$.*

Proof. Suppose E is naa-preferred. Then $ad^{naa}(F^E) = \{\emptyset\}$ because if some nonempty $D \in ad^{naa}(F^E)$ existed, $E \cup D \in ad^{naa}(F)$ by modularization, so E would not be maximal. Now if there are no naa-admissible arguments in the reduct of E, then E defeats all arguments in F^E, so $A = E \cup F^E \cup E^+ = E \cup Def_{naa}(E)$, thus E is naa-stable. For the other direction this argumentation can be reversed. □

3.4 Admissibility under undecidedness blocking

At first glance it looks like lub-semantics are not concerned with defense at all, in Prop. 2.40 we show that instead of defending itself an lub-complete extension only needs to contain the arguments it c-defends. And yet a certain basic principle of defense is still at work here. By closer inspection none of the lub-complete extensions are attacked by unattacked arguments. This is a consequence of unattacked arguments being defended by any extension, so being attacked by them would result in an inner conflict. It seems the combination of c-completeness with conflictfreeness is enough to make some attackers, like unattacked arguments, undefeatable for lub-complete extensions. The following example outlines another type of attacker that cannot be ignored.

Example 3.19. In Example 2.55 $\{c\}$ is not an lub-complete extension because it does not contain e which is c-defended by c. The reason why e is a problematic attacker is easier to see if one tries to construct an lub-complete labeling. If we set $lab(c) = in$, $lab(d) = out$ is inevitable and as a consequence $lab(e) = in$ which leads to a conflict among in-arguments. b on the other hand, the other attacker of c, causes no such problem, because it is attacked by a so both can simply be labeled undecided. The difference between the two attackers seems to be that e is unattacked in $F^{\{c\}}$ while b is not.

The reduct is our best option to identify both types of problematic attackers. If an argument is unattacked in the reduct F^E of an extension E then either it is unattacked in the original AF, too, or all its attackers are in turn attacked by E, meaning it is c-defended by E. If such an argument attacks E, E cannot be lub-complete. But excepting those unattacked arguments of the reduct from being defeated is not enough, because they in turn can c-defend arguments which also have to be included for lub-completeness and can therefore not be defeated. This closure of unattacked arguments under c-defense is known as the c-grounded extension, in this case the c-grounded extension of the reduct. From the information which types of attackers are not defeated by lub-complete extensions we can deduce which are actually defeated - the rest. The defeat operator for lub-semantics we arrived at is thus:

Definition 3.20. Let $F = (A, R)$ be an AF. We define the lub-defeat operator Def_{lub} for any $E \subseteq A$ to be $Def_{lub}(E) \coloneqq \{a \in A \mid a \notin gr^c(F^E) \cup E\}$.[1]

[1]Remember that there exists exactly one c-grounded extension, so we use $a \notin gr^c(F^E) \cup E$ as a short version for $a \notin G \cup E$ with $G \in gr^c(F^E) = \{G\}$

Note that this defeat operator is not recursive. To determine the defeated arguments the reduct and its c-grounded extension have to be computed only once. We will now prove that the lub-complete extensions are indeed the fixpoints of the defense operator induced by Def_{lub}.

Proposition 3.21. *Let $F = (A, R)$ be an AF and $E \subseteq A$. E is an lub-complete extension of F iff $E = \chi_{lub}(E)$.*

Proof. (1) Suppose E is lub-complete. By definition E is conflictfree. Since $\Gamma(E) \subseteq E$, E contains all arguments unattacked in $A \setminus E^+$, so $gr^c(F^E) = \emptyset$. It follows that all arguments not in E are in $Def_{lub}(E)$. Therefore $Free_{lub}(E) = \chi_{lub}(E) = E$. Suppose now $E = \chi_{lub}(E)$ then E is conflictfree. We know $\Gamma(E) \setminus E$ is unattacked in the reduct and therefore contained in its grounded extension. Since all attackers of $\Gamma(E)$ are attacked by E, $\Gamma(E)$ is unattacked in $Free_{lub}(E)$, so $\Gamma(E) \subseteq \chi_{lub}(E) = E$. □

The existence of a defeat operator which has the lub-complete semantics as its induced complete semantics makes it possible to define an admissible version of lub-semantics.

Definition 3.22 (lub-admissible)**.** Let $F = (A, R)$ be an AF and $E \subseteq A$.
E is *lub-admissible* iff $E \subseteq \chi_{lub}(E)$.

By defining an lub-admissible semantics the defense behavior of lub-semantics is successfully isolated and the completeness part of the original semantics can fulfill its intended purpose of realizing the reinstatement property. We are now able to differentiate between an argument set that is not lub-complete because it is missing some arguments (but is still lub-admissible) and one that is neither lub-complete nor lub-admissible because it has an attacker it cannot lub-defend itself against. Take the following example for this.

Example 3.23. In Example 2.2 the singleton $\{c\}$ is an lub-admissible extension that is not lub-complete. The reduct $F^{\{c\}}$ consists of arguments a and d. Since d is a self-attacker but a is unattacked, the c-grounded extension of the reduct is $\{a\}$, so d and b, which is attacked by c and is therefore also no member of $gr^c(F^{\{c\}})$, are defeated. The remaining arguments c and a are both unattacked, so $\{c\} \subset \chi_{lub}(\{c\})$ is lub-admissible but not a fixpoint, so it is not lub-complete.

Like their classic counterparts, lub-admissible extensions can be completed. In contrast to them only one step is needed here.

Proposition 3.24. *Let $F = (A, R)$ be an AF and $E \subseteq A$ lub-admissible. Then $\chi_{lub}(E)$ is an lub-complete extension containing E.*

Proof. If E is lub-admissible then it is contained in $\chi_{lub}(E)$ by definition. Because E is lub-admissible it is known to be unattacked in $Free_{lub}(E) = E \cup gr^c(F^E))$, thus E is conflictfree. By definition the c-grounded extension of F^E is conflictfree, too, and (since E is unattacked in $Free_{lub}(E)$) it does not attack E, so $Free_{lub}(E) = \chi_{lub}(E) = E \cup gr^c(F^E)$ is conflictfree.
According to Prop. 2.40 it is only left to show that $\Gamma(\chi_{lub}(E)) \subseteq \chi_{lub}(E)$. Now let $a \in \Gamma(\chi_{lub}(E))$, then $a \notin E^+$, because $\chi_{lub}(E)$ is conflictfree. If $a \in E$ it is also in $\chi_{lub}(E)$ because E is lub-admissible. Suppose now $a \in F^E$. Since a is c-defended by $\chi_{lub}(E)$ every attacker y of a is either attacked by E, and thus $y \notin F^E$ or attacked by some $g \in gr^c(F^E)$ but then by the c-completeness of the c-grounded semantics a is a member of $gr^c(F^E)$, so $a \in gr^c(F^E) \subseteq \chi_{lub}(E)$. □

We now conduct a short, informal principle-based analysis of the lub-admissible semantics with the principles from [Dondio and Longo, 2021]. ad^{lub} satisfies conflictfreeness and, since it is extension-based, rejection. It probably satisfies directionality and abstention, we leave the proof for that for future work. It does not satisfy naivity, admissibility, I-maximality or cycle-homogeneity.[2] Reinstatement is a property reserved for complete semantics and is not satified by the lub-admissible semantics (on purpose), making this the only notable difference between it and the rest of the family regarding the criteria from [Dondio and Longo, 2021].[3]

Since any lub-admissible extension has an lub-complete extension for a superset, the lub-preferred semantics introduced in Section 2.3.2 coincides with the preferred semantics induced by Def_{lub} that is the set of maximal lub-admissible extensions. This completes the embedding of the lub-admissible semantics in the lub-semantics family.

Corollary 3.25. $pref^{lub}(F) = \{E \in ad^{lub}(F) \mid E \subseteq -maximal\ in\ ad^{lub}(F)\}$ *holds for any AF $F \in U_F$.*

Proof. Follows directly from Prop. 3.24 □

[2] For those properties that are not satisfied see [Dondio and Longo, 2021], as every lub-complete extension is lub-admissible

[3] At least at the moment, as mentioned, some proofs are still missing

3.5 A new defeat-based recursive semantics

Naa-defense is significantly stricter than lub-defense, which can already be guessed from the resp. defeat operators. While Def_{lub} defeats every argument with a controversial status in the reduct, Def_{naa} only defeats arguments that are not naa-admissible in the reduct. The following example demonstrates this difference.

Example 3.26. In Example 2.22 c is lub-admissible, because its attackers are in conflict with each other and therefore not c-grounded, but not naa-admissible, because both attackers defend themselves against each other and thus are naa-admissible in the reduct.

One can argue naa-defense is too strict in cases like this. On the other hand there are cases like Example 2.41 in which lub-defense is too liberal. As we said in Section 2.3.2 we will now propose a reduct-based alternative to the SCC-recursive ub-semantics in [Dondio and Longo, 2021] which produces the desired outcomes for both cases.

With the general defense notion introduced in Section 3.2 all we have to do for that is to choose the right defeat operator. So the question is which attackers should be defeated under our semantics and which should not. Of course all arguments defeated under naa-semantics should be defeated by the new operator, too, which means we have to include the defeat conditions from Def. 3.16 and thus inherit recursiveness. Additionally we would want to defeat conflicting attackers like a and b from Example 2.22. What we want to avoid is defeating arguments with only non-admissible attackers, like d from Example 2.41. This can be summarized as a principle of *no uncontroversial attackers (nua)* and is accomplished by defeating only those admissible arguments where every extension containing them is under attack by another admissible extension. Our new defeat operator therefore has the following three defeat conditions.

Definition 3.27 (nua-defeat operator)**.** Let $F = (A, R)$ be an AF and $E \subseteq A$. We define $Def_{nua}(E)$ to be the set of all $a \in A$ that satisfy one of the following conditions:

1. $a \in E^+$ (a is attacked by E)
2. $a \notin ad^{nua}(F^E)$ (a is not part of any nua-admissible extension in the reduct)
3. $\forall D \in ad^{nua}(F^E), a \in D\ \exists C \in ad^{nua}(F^E)\ \exists c \in C : c \to D$ (for every nua-admissible extension containing a in the reduct exists another extension in the reduct that attacks it)

Like with naa-defeat and -admissibility, nua-defeat is defined recursively over the reduct. So this definition presupposes the term nua-admissibility which is defined below. It is well-defined, since the reduct is a proper subset for any $E \neq \emptyset$ and the empty set is nua-admissible trivially. The corresponding defense operator χ_{nua} to Def_{nua} is used to generate the nua-semantics family.

Definition 3.28 (nua-semantics family)**.** Let $F = (A, R)$ be an AF and $E \subseteq A$.

E is nua-admissible iff $E \subseteq \chi_{nua}(E)$.

E is nua-complete iff $E = \chi_{nua}(E)$.

E is nua-preferred iff E is maximal w.r.t. $\subseteq$ in $ad^{nua}(E)$.

E is nua-grounded iff E is minimal w.r.t. $\subseteq$ in $co^{nua}(E)$.

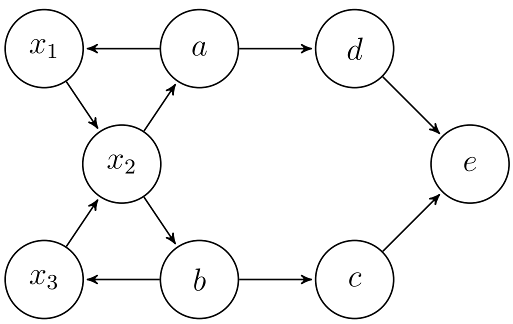
The reader may convince herself that Example 2.22 is handled by the different nua-semantics in the same way as by the resp. lub-semantics. We will apply nua-semantics to the other motivating example here instead.

Example 3.29. Example 2.41 continued. The set of all nua-admissible extensions is $ad^{nua}(F) = \{\emptyset, \{d\}\}$ with $\{d\}$ being the only nua-preferred,-complete and -grounded extension.

The nua-semantics family solves all our motivational examples but apart from that these semantics are missing some important properties. They are the first defeat-based semantics family where the preferred extensions are not necessarily complete which proves that this coincidence is not a property of defeat-based semantics in general.

Proposition 3.30. *There exists an AF $F = (A, R)$ such that not every nua-preferred extension of F is nua-complete.*

Counterexample 3.31. For the AF below $\{e\}$ is nua-preferred but not nua-complete, because $\chi_{nua}(\{e\}) = \{a, b, e\}$.



This feature of nua-semantics leads to some difficulties. The existence of nua-complete and nua-grounded extensions has become an open question. In order to investigate the performance of our new semantics family more thoroughly we conduct a principle-based analysis. For comparison we use the criteria from [Baumann et al., 2020a].

Proposition 3.32 (principle-based analysis of nua-semantics)**.**

I. *Naivety, c-admissibility and strong admissibility are not satisfied by any of the four nua-semantics.*

II. *Reinstatement is satisfied by* co^{nua}, gr^{nua} *and* $pref^{nua}$ *but not by* ad^{nua}.

III. *I-maximality is satisfied by* gr^{nua} *and* $pref^{nua}$ *but not by* co^{nua} *and* ad^{nua}.

IV. *Directionality is satisfied by* ad^{nua} *but not by* $pref^{nua}$.

V. *Modularization and meaningless reduct are not satisfied by any nua-semantics.*

VI. *Unattack inclusion is satisfied by* ad^{nua} *and* $pref^{nua}$.

VII. gr^{nua} *is not unique.*

Proof. (I), (V) Example 2.22 with the extension $\{c\}$ can serve as a counterexample for ad^{nua}, $pref^{nua}$ and co^{nua}. For gr^{nua} consider Counterexample 3.31 which has the two nua-grounded extensions: $\{a, c\}$ and $\{b, d\}$(it therefore also proves (VII)).
(II) Since $E^+ \subseteq Def_{nua}(E)$ any c-defended argument a is unattacked in F^E and thus $a \in \chi_{nua}(E)$. So the fixpoint semantics co^{nua} and gr^{nua} have to include such an a in their extensions. For $pref^{nua}$ note that any argument b attacked by such an a in the reduct is not nua-admissible, so $E \cup a$ is nua-admissible, because only ineffective attackers are erased from F^E.
(III) Follows directly from the minimality resp. maximality w.r.t. set inclusion.
(IV) Proof by induction over size of F, trivial base case. Suppose $U \subseteq A$ unattacked, then for any nonempty $E \in ad^{nua}(U)$ the reduct F^E satisfies the induction hypothesis, so all nua-admissible arguments of U^E are nua-admissible in F^E and the other way around. Therefore E still defeats all of its attackers from U^E in F^E and thus E is nua-admissible in F. For the other direction we can argue again that every nua-admissible extension attacking $E \cap U$ in U^E has an nua-admissible counterpart in F^E that attacks E by the induction hypothesis. The only thing nontrivial here is the question if any extension containing such an attacker is always attacked by another nua-admissible extension *in U*. But in the first part of the proof we have already shown that an nua-admissible extension of an unattacked set has not only a counterpart in F but is itself nua-admissible in F, so the attack that defeats it and makes E nua-admissible in F must come from

an argument of the unattacked U^E.
For $pref^{nua}$ the nua-preferred extension $\{c\}$ of Example 2.22 can serve as a counterexample, because its intersection with the unattacked set $U = \{a, b\}$ is empty and not nua-preferred in U because $\{a\}$ is an nua-admissible superset of the empty set in U.
(VI) The set of all unattacked arguments is always nua-admissible and thus has an nua-preferred superset.

□

The loss of the modularization property is an unavoidable consequence of further weakening naa-defeat according to Theorem 4.4 from [Baumann et al., 2020a]. The nua-admissible semantics now serves as a "living" example for this. How much of a difference modularization makes will be discussed further in Chapter 5. A number of results are still missing here, mainly because there is no proof for the existence of nua-complete and nua-grounded extensions in general yet.

Conjecture 3.33. *I. $co^{nua}(F)$ is not empty for any AF $F \in U_F$.*

II. Unattack inclusion is satisfied by co^{nua} and gr^{nua}.

III. Directionality is satisfied by co^{nua} and gr^{nua}.

Nua-defense was meant to be a compromise between the strict naa-defense and the liberal lub-defense. However, the nua-complete semantics is not well understood yet and at this point it is already clear that the nua-semantics family is missing an important property the other two have.[4] The benefits from having a second recursive semantics family comparable to naa-semantics have made the construction of nua-semantics worthwhile though. To sum up this section, it is easy to construct a new semantics family by modifying an existing defeat operator but not that easy to guarantee specific properties are still satisfied by the induced semantics.

[4]Namely the coincidence of preferred with maximal complete extensions

3.6 Neutralizing self-attackers with cogent defeat

The relative defense condition of cogent semantics seems to require the defeat of sets of arguments instead of single arguments. But the cogent semantics is not as complicated as it looks. Before we give the defeat operator for it, we will first prove that Def. 2.34 is just an interesting way of adding self-attackers to the set of attackers that can be ignored by an extension.

Proposition 3.34. *Let $F = (A, R)$ be an AF and $E \subseteq A$. Then E is cogent iff E is conflictfree and* $\forall\, a \in E^- :\ a \in E^+ \ \vee\ a \to a$

Proof. ($\Rightarrow$) Suppose E is cogent. Then E has to be conflictfree or it would not be c-admissible in $F\downarrow_{E\cup\emptyset}$. For any attacker a of E we have to consider $F\downarrow_{E\cup\{a\}}$. If a is a self-attacker, then $\{a\}$ is not c-admissible and thus not at least as cogent as E.[5] If not E has to be c-admissible in $F\downarrow_{E\cup\{a\}}$ so E attacks a.
($\Leftarrow$) We need to show that such an E is at least as cogent as any $D \subseteq A$ that is at least as cogent as E. If D is not conflictfree, it is not c-admissible in $F\downarrow_{E\cup D}$ and therefore not at least as cogent as E. If D is conflictfree, then it contains no self-attackers, so any attacker d of E in D is attacked by E. But then, since E is conflictfree, E is c-admissible in $F\downarrow_{E\cup D}$ and thus at least as cogent as D. □

This proof demonstrates that, while cogent semantics makes modifications to c-semantics on extension level, the relevant mechanisms can still be broken down to the behavior of single arguments. It is now possible to express the cogent semantics in terms of our general defense notion. The defeat operator is easily derived from Prop 3.34 - it defeats all attacked arguments and all self-attackers.

Definition 3.35 (cogent defeat)**.** Let $F = (A, R)$ be an AF. Define the cogent defeat operator Def_{cog} for any $E \subseteq A$ as $Def_{cog}(E) \coloneqq \{a \in A \mid E \to a \ \vee\ a \to a\}$ the set of all arguments that are either self-attackers or attacked by E.

Proposition 3.36. *For any AF $F = (A, R)$ and any $E \subseteq A$ it holds that E is cogent iff $E \subseteq \chi_{cog}(E)$ for the defense operator χ_{cog} induced by the cogent defeat operator Def_{cog}.*

Proof. Follows directly from Prop.3.34. □

[5] and the same applies to any set of arguments D containing such an a

If the admissible extensions of Def_{cog} coincide with the original cogent extensions, the cog-preferred semantics *is* the sustainable semantics. We have thus succeeded in describing all three weak semantics families from Section 2.3 with our general defense notion.

Corollary 3.37. *Let $F = (A, R)$ be an AF. Then the set of all cogent-preferred extensions $pref^{cog}(F)$ coincides with the set of all sustainable extensions of F.*

With the cogent defeat operator known we can expand the cogent semantics family by adding the corresponding complete and grounded semantics.

Definition 3.38. Let $F = (A, R)$ be an AF. The sets of cog-complete and cog-grounded extensions on F are given by $co^{cog}(F) = \{E \in ad^{cog}(F) \mid \chi_{cog}(E) = E\}$ and $gr^{cog}(F) = \{E \in co^{cog}(F) \mid E \subseteq -minimal\ in\ co^{cog}(F)\}$ respectively.

For a quick demonstration of these semantics consider our first running example again.

Example 3.39. In Example 2.2 $\{a, c\}$ is the only cog-complete and cog-grounded extension, because the empty set and $\{c\}$ both cog-defend a and a cog-defends c.

We will leave the conduction of a principle-based analysis of these two semantics for future work and only include a proof for the coincidence of maximal cog-complete extensions with sustainable (cog-preferred) extensions here.

Proposition 3.40. *Let $F = (A, R)$ be an AF. Then*

$$pref^{cog}(F) = \{E \in co^{cog}(F) \mid E \subseteq -maximal\ in\ co^{cog}(F)\}$$

Proof. Follows from Prop. 4.21 and Cor. 4.25. The reasoning is that $\chi_{cog}(E)$ is cog-admissible if E itself is cog-admissible, because none of the defended arguments attack each other. □

In conclusion it can be said that the cogent semantics family is very close to Dungs semantics, probably the closest among all weak semantics considered here. While ignoring self-attackers might seem simple compared to e.g. naa-semantics, the cogent semantics are nonetheless a valuable and well designed option to refine classic admissibility with close to no deficits when it comes to classic behavior. In the next chapter, where the properties of the classic defense operator are examined for the general case, this will become even more obvious.

Chapter 4

Analyzing semantic properties via defeat operators

4.1 Comparing defeat operators

One important factor in our motivation for a unified defense notion was the formalization of a *weaker-as*-relation between the defense concepts of different semantics.[1] With cogent semantics having only an admissible, lub-semantics only a complete, and naa-semantics having a complete semantics which seemed very far apart from its admissible semantics, the standard way of checking for containment of each other was insufficient for this. By isolating the defense behavior in the defeat operators and generalizing the construction of a semantics family we managed to close these gaps and can now directly compare defeat operators instead of their induced semantics. Nonetheless, we decided to start our comparison with containment between the different admissible semantics for a smooth transition from the standard approach and the inclusion of known results.

Definition 4.1 (weakness relation)**.** Let δ, ϵ be defeat operators. δ is *at least as weak* as ϵ, $\delta \leq_w \epsilon$ iff $ad^\epsilon(F) \subseteq ad^\delta(F)$ for all $F \in U_F$.
δ is *strictly weaker* than ϵ, $\delta <_w \epsilon$ iff δ is at least as weak as ϵ and there exists some $F = (A, R) \in U_F,\ E \subseteq A$ such that $E \in ad^\delta(F)$ but $E \notin ad^\epsilon(F)$.

The admissible semantics is the generic type of defeat-based semantics, while the other semantics are subsets of their resp. admissible semantics. One can thus argue the admissible semantics exhibits the defense behavior under a certain defeat

[1] A more general proposal for the comparison of argumentation semantics can be found in [Amgoud and Prade, 2009]

operator in its purest form. The resp. complete semantics would be another viable choice for further investigations.

Proposition 4.2 (weakenings of c-admissibility)**.** *Def_δ is strictly weaker than Def_c for $\delta \in \{naa, lub, nua, cog\}$.*

This was already noted for naa-semantics in [Baumann et al., 2020b], for cogent semantics it follows directly from Prop. 3 and Def. 4 in [Bodanza and Tohmé, 2009]. For the other two semantics we will give the proof after the introduction of a direct relation between defeat operators.

Definition 4.3 (aggressiveness)**.** Let δ, ϵ be defeat operators. δ is *at least as aggressive* as ϵ, $\delta \leq_a \epsilon$ iff $\epsilon(E) \subseteq \delta(E)$ for all $E \in A,\ F = (A, R) \in U_F$.
δ is *strictly more aggressive* than ϵ, $\delta <_a \epsilon$ iff it is at least as aggressive as ϵ and some E with $\delta(E) \setminus \epsilon(E) \neq \emptyset$ exists.

Both aggressiveness and weakness are partial-orders on the set of all defeat operators (since they are based on set-inclusion, which is a partial order itself). The set of all defeat operators together with aggressiveness even forms a complete lattice, e.g. the greatest lower bound of two defeat operators is $\delta \vee \epsilon\ (F, E) := \delta(F, E) \cup \epsilon(F, E)$ the least aggressive defeat operator which is at least as aggressive as δ and as ϵ. Being more aggressive does not imply being weaker, so the completeness under the $\leq_w$-relation is an open question at this point. For example, the most aggressive defeat operator always defeats everything, which makes the empty set its only admissible extension. So while on the one hand this operator is strictly more aggressive than Def_c, Def_c is strictly weaker than it on the other hand. In order to accept more arguments we therefore cannot simply make defeat more aggressive. The following property solves this problem and will come in handy in other contexts, too.

Definition 4.4 (selfdefeatfree)**.** A defeat operator δ is *selfdefeatfree* iff for any AF $F = (A, R)$ and any $E \in cf(F)$ it holds that $E \cap \delta(E) = \emptyset$.
δ is *strictly selfdefeatfree* iff $E \cap \delta(E) = \emptyset$ for any $E \subseteq A$.

As already mentioned in the context of Theorem 3.10 a weak semantics should ideally only accept selfdefeatfree extensions. For the weak semantics introduced her this was realized by making their defeat concept selfdefeatfree to begin with. The reduct is one way to hardcode the exclusion of selfdefeat, the defeat operator for lub-semantics another. With Theorem 3.10 defeat operators which are not trivially selfdefeatfree have become a safe option for designing weak semantics. This widens the possibilities for choosing efficient defeat criteria without the risk of selfdefeating extensions.

Proposition 4.5. *Def_{cf} and Def_{lub} are strictly selfdefeatfree. Def_δ is selfdefeatfree for any $\delta \in \{c, naa, nua, cog\}$.*

The proof is omitted, the statements can be deduced from the respective definitions of the defeat operators with reasonable effort. For selfdefeatfree operators aggressiveness coincides with weakness, because they only defeat potential threats for some E while E itself is guaranteed to be included in $Free_\delta(E)$. Note that it is enough for the more aggressive operator to be selfdefeatfree in order to be weaker because an operator which is less aggressive than a selfdefeatfree operator has to be selfdefeatfree, too.

Proposition 4.6. *Let δ, ϵ be defeat operators. If δ is selfdefeatfree and at least as aggressive as ϵ then δ is at least as weak as ϵ.*

Proof. Let $F = (A, R)$ and $E \subseteq A$. If $E \in ad^\epsilon$, E is conflictfree. Since δ is self-defeat free, $E \subseteq Free_\delta(E)$ and since δ is at least as aggressive as ϵ it follows that $Free_\delta(E) \subseteq Free_\epsilon(E)$. Therefore if E is unattacked in $Free_\epsilon(E)$ it is also unattacked in $Free_\delta(E)$ so $E \subseteq \chi_\delta$. □

With Def_{cf} being the most aggressive strictly selfdefeatfree defeat operator we now have a formal account for the statement that the conflictfree semantics ad^{cf} has the weakest defense concept. The minimal weakness of Def_{cf} also follows directly from the conflictfreeness of admissible extensions in general (Theorem 3.10). The c-admissible semantics on the other hand has been shown to be weaker than total defeat. Classic defense is therefore just one defense concept among others. The nonetheless important containment of ad^c in ad^{nua} and ad^{lub} can be shown as a consequence of Prop 4.6.

Proof for weakenings of c-admissibility. Since both Def_{lub} and Def_{nua} are selfdefeatfree, it follows from Prop. 4.6 that Def_{nua} and Def_{lub} are at least as weak as Def_c because $Def_c(E) = E^+ \subseteq Def_{nua}$ for any $E \subseteq A$ of any $F = (A, R) \in U_F$. In case of Def_{lub} we can consider the strictly selfdefeatfree version of classic defeat instead, $Def_{c'}(E) = E^+ \setminus E$. For any conflictfree E we have $Def_{c'}(E) = Def_c$ and since admissible extension are always conflictfree it follows that $ad^{c'}(F) = ad^c(F)$ for any $F \in U_F$. But for $Def_{c'}$ it clearly holds that $Def_{c'}(E) = (E^+ \setminus E) \nsubseteq (gr^c(F^E) \cup E)$ so $Def_{c'}$ is at least as weak as Def_{lub} by Prop. 4.6 and thus $ad^c(F) \subseteq ad^{lub}(F)$ for all $F \in U_F$.
Both semantics are strictly weaker because for $F = (\{a, b\}, \{(a, a), (a, b)\})$ we have $\{b\} \in ad^{lub}(F)$ and $ad^{nua}(F)$ but not in $ad^c(F)$. □

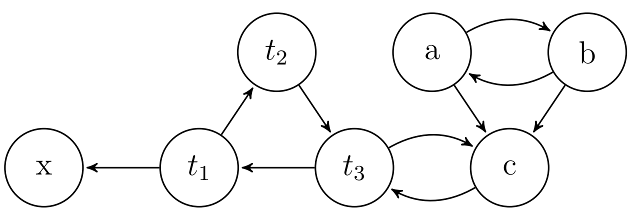
We conclude this section by providing the $<_w$-order among the defeat operators from Chapter 3.

Theorem 4.7. $Def_{cf} <_w Def_{lub} <_w Def_{nua}, Def_{naa} <_w Def_{cog} <_w Def_c$

Proof. With Prop. 4.5 this amounts to comparing aggressiveness. All operators defeat E^+(at least for conflictfree E), so we can focus on the defeat within the reduct. $Def_{cf} \leq_w Def_{lub}$ is trivial, because all defeat-based semantics are conflictfree. Self-defeaters are not conflictfree and therefore not naa-admissible so $Def_{cog}(E) \subseteq Def_{naa}(E)$ for all $E \in cf(F),\ F = (A, R) \in U_F$, the same applies to Def_{nua}. $Def_{cog} \leq_w Def_c$ was already shown in Prop. 4.2.
($Def_{lub} \leq_a Def_{nua}$) The c-grounded extension of the reduct is always nua-admissible and the arguments attacked by the c-grounded extension cannot be nua-admissible because the c-grounded extension is recursively constructed from unattacked arguments. Therefore no attacker of $gr^c(F^E)$ could nua-defend itself in the reduct, so the arguments of $gr^c(F^E)$ are never nua-defeated. Since Def_{lub} defeats everything except for $gr^c(F^E)$, it is thus at least as aggressive as Def_{nua}. A similar argument can be used to show $Def_{lub} \leq_a Def_{naa}$.
Example 2.41 can serve for the strictly weaker part in the first 3 cases, Example 2.2 for $Def_{cog} <_w Def_c$. □

As the following counterexample shows, the two recursive defeat operators Def_{naa} and Def_{nua} are incomparable. On the one hand the singleton $\{c\}$ is nua-admissible but not naa-admissible. On the other hand the singleton $\{x\}$ is naa-admissible because both $\{t_1\}$ and $\{t_1, c\}$ have attackers they cannot naa-defeat in $F^{\{x\}}$, so x has no naa-admissible attacker. Under nua-defeat, however, the singleton $\{t_1\}$ can defeat t_3, because t_3 is attacked by the nua-admissible c in $F^{\{x\}^{\{t_1\}}}$, so t_1 is a nua-admissible attacker. Its only attacker t_3 is not nua-admissible because it cannot be defended against t_2. Therefore t_1 is not nua-defeated by $\{x\}$ so $\{x\}$ is not nua-admissible.

Counterexample 4.8.



4.2 Monotonicity

The following sections each examine a property of the Γ-Operator for the general case - monotonicity, the Fundamental Lemma and additivity of defeat. We start with monotonicity because the monotonicity of a defense operator implies the generalized Fundamental Lemma under our defense notion (since defended sets are always conflictfree). Monotonicity originates from the mathematical field of Analysis. In the context of lattice theory it is defined for any function on a poset[2] as the following property.

Definition 4.9. Let $(P, \leq)$ be a poset. A unary Operator $op : P \to P$ on P is monotonic iff $\forall\, x, y \in P: \; x \leq y \Rightarrow op(x) \leq op(y)$[3]

The poset in question is in our case $(2^A, \subseteq)$, the powerset of the set of arguments for an AF $F = (A, R)$ with set inclusion as a partial order. We already noted that the Γ-Operator is monotonic on this poset for any F. Other examples include some of our defeat operators.

Example 4.10. Let $F = (A, R)$ be an AF. Then Def_c and Def_{cog} are monotonic on $(2^A, \subseteq)$.

Monotonic defeat operators are interesting but not exactly delivering when it comes to the properties of the resulting semantics. For one thing monotonicity tends to get in conflict with being selfdefeatfree. The bigger the argument set the less options for structurally motivated defeat are left. Monotonicity backwardly limits the defeat options for all subsets of such a big argument set which is why it only makes sense for stronger semantics to be monotonic. Lub-defeat for example allows extensions to defeat arguments they could also take in - a direct clash between monotonicity and selfdefeatfreeness. However, monotonicity is highly relevant when it comes to the defense operators. The reason behind this is the Knaster-Tarski-Theorem which guarantees the existence and uniqueness of both a minimal and a maximal fixpoint for monotonic operators on complete lattices[4] and with set inclusion as our partial order completeness is given for $(2^A, \subseteq)$ with the standard cut and union operations on sets as the upper and lower bounds.

Theorem 4.11 (Knaster-Tarski-Theorem)**.** *For any monotonic $op : P \to P$ on a complete lattice $(P, \leq)$ the set of all fixpoints $F_{op}(P) := \{p \in P \mid p = op(p)\}$ is a complete lattice, too.*[5]

[2] more generally also between posets
[3] [Tarski, 1955]
[4] A complete lattice is a poset where every subset has a least upper and greatest lower bound
[5] This theorem has been proven by Tarski in [Tarski, 1955]

So under a monotonic defense operator the existence and uniqueness of a grounded extension is guaranteed, like for the classic Γ-Operator, and by implication the existence of complete extensions in general is assured. Moreover, a monotonic defense operator also yields a unique maximal fixpoint because, unlike the classic approach, conflictfreeness is not a seperate criterion under the new defense notion. We will therefore now examine under which conditions a defense operator is monotonic.

Proposition 4.12. *Let $F = (A, R)$ be an AF and let δ be an arbitrary defeat operator. Then χ_δ is monotonic iff for any $E, E' \in 2^A$ with $E' \subseteq E$:*

1. $(\chi_\delta(E'))^- \subseteq \delta(E)$.
2. $\delta(E) \cap \chi_\delta(E') = \emptyset$

We want E to defend $\chi_\delta(E')$ so $\chi_\delta(E')$ has to stay unattacked in $Free_\delta(E)$. Towards this end two conditions have to be fulfilled, (2) guarantees $\chi_\delta(E')$ is contained in $Free_\delta(E)$ and (1) that it is unattacked. On the defeat level these conditions come down to the following:

Proposition 4.13. *Let $F = (A, R)$ be an AF and let δ be an arbitrary defeat operator. Then χ_δ is monotonic iff for any $E, E' \in 2^A$ with $E' \subseteq E$ and for any argument $a \in A$: If $\{a\}^- \subseteq \delta(E') \wedge a \notin \delta(E')$ then $\{a\}^- \subseteq \delta(E) \wedge a \notin \delta(E)$*

Proof. Both propositions follow from $\chi_\delta(E') = \{a \in A \mid \{a\}^- \subseteq \delta(E') \wedge a \notin \delta(E')\}$. □

As the proof says, we need to protect exactly those arguments from defeat through E which are not defeated by E' while having all their attackers defeated by it. Note that δ does not have to be monotonic and that at the same time monotonicity of δ is not sufficient for this. For example χ_c is not monotonic because the conflictfree subsets of a set E may defend themselves while the set E might end up defeating them e.g. in the case $E = \{a, b\}$ in Example 2.22. As just demonstrated the classic defense operator under the new defense notion is no longer monotonic and classic defense is no special case in this regard. It turns out monotonicity of χ_δ imposes a strong limitation on the resulting semantics in general, namely:

Theorem 4.14. *Let $F = (A, R)$ be an AF and let δ be a defeat operator satisfying $\chi_\delta(E') \subseteq \chi_\delta(E)$ for all $E, E' \in 2^A$ with $E' \subseteq E$ for the corresponding defense operator. Then $\bigcup_{E \in ad^\delta(F)} E$ is conflictfree.*

Proof. By Theorem 3.10 we know that $\chi_\delta(A)$ is always conflictfree. Now for any extension $E \in ad^\delta(F)$ it holds that, $E \subseteq A$, so $\chi_\delta(E) \subseteq \chi_\delta(A)$ and since E is admissible we have $E \subseteq \chi_\delta(E) \subseteq \chi_\delta(A)$. So all admissible extensions are subsets of the conflictfree $\chi_\delta(A)$ and thus their union is, too. □

In short the set of credulously accepted arguments under a monotonic defense operater χ_δ is always conflictfree, which also means no two extensions under χ_δ are in conflict with each other. If that holds it is no longer surprising to have a unique maximal complete extension. But the resulting semantics end up being trivial. Semantics induced by constant defeat operators for example, like the following.

Example 4.15. The empty defeat operator $Def_\emptyset(F, E) = \emptyset$ induces a constant and thus monotonic defense operator, $\chi_\emptyset(E) = \{a \in A \mid a\ unattacked\}$ the set of all unattacked arguments for any $E \in A,\ F = (A, R) \in U_F$. This set is then the only complete, preferred and grounded extension.

So monotonicity of the defense operator leaves no room for ambiguity in complete extensions (and also not in admissible extensions). A consequence of this is that the Knaster-Tarski Fixpoint-Theorem does not apply to any defense operator that yields conflicting extensions under our defense notion which includes the classic semantics and all four weak semantics investigated here. To obtain similar results to Knaster-Tarski for semantics which allow conflicting extensions it makes sense to ask for a monotonic $Free_\delta$.

Proposition 4.16. *Let $F = (A, R)$ be an AF and let δ be an arbitrary Defeat Operator. Then $Free_\delta$ is monotonic iff δ is antimonotonic i.e. for all $E, E' \in 2^A$ with $E' \subseteq E$ it holds that $\delta(E) \subseteq \delta(E')$.*

Proof. Follows directly from the definition of $Free_\delta(E) \coloneqq A \setminus \delta(E)$. □

Antimonotonic defeat mashes well with selfdefeatfree operators but not so well with classic defeat which is included in one form or another in all four weak semantics considered in this work (see the counterexample below). So far the only reasonable example for an antimonotonic defeat operator is the cf-semantics.

Example 4.17. $Free_{cf}$ is monotonic because $Def_{cf}(E) = A \setminus E$, the cf-defeat operator, is clearly antimonotonic.

Proposition 4.18. *$Free_\delta$ is not monotonic for $\delta \in \{naa, lub, nua, cog\}$*

Counterexample 4.19. Example 2.2 is a counterexample for this.
$Free_\delta(b) = \{a, b\} \not\supseteq \{a, c\} = Free_\delta(\emptyset)$ for $\delta \in naa, lub, nua$. For cogent semantics we even have $Free_{cog}(\emptyset) = \{a, b, c\}$.

Conclusively one can say that monotonicity is too restrictive in combination with our defeat notion to be a valuable property for any of the operators involved. We will therefore investigate another option to guarantee the existence of complete extensions in the next section - the Fundamental Lemma.

4.3 Generalizing the Fundamental Lemma

If lattice theory cannot guarantee the existence of complete extensions, an iteration algorithm is our alternative of first choice. In the classic case the Fundamental Lemma (Prop. 2.23) is the foundation of iterating the Γ-Operator on admissible extensions. It can be generalized to the following property for defeat-based defense operators:

Definition 4.20 (General Fundamental Lemma)**.** A defeat operator δ satisfies the generalized Fundamental Lemma iff $E \subseteq E' \subseteq \chi_\delta(E)$ implies $\chi_\delta(E) \subseteq \chi_\delta(E')$ for all $E, E' \subseteq A,\ F = (A, R) \in U_F$.

The conclusion of our generalized Fundamental Lemma is actually twofold. Obviously all arguments defended by E are still defended by E'. Indirectly, that includes E' being defended by E' so the δ-admissibility of E' is also guaranteed by the conclusion. These conclusions correspond to the two conclusions of the original lemma, both admissibility and defended arguments are maintained under adding defended arguments to an extension. Integrating conflictfreeness in our defense notion has again simplified things. Once a fixpoint is reached by iterating the defense operator that fixpoint is guaranteed to be a complete extension even for non-classic defeat. This is not trivial because under an arbitrary defeat notion conflictfreeness of E' cannot be reduced to the conflictfreeness of E like in the classic case where attack is defeat. The generalized Fundamental Lemma divides the semantics introduced in two groups. The non-recursive semantics satisfy it, the recursive do not.

Proposition 4.21. *Def_c, Def_{cog} and Def_{lub} satisfy the generalized Fundamental Lemma, while Def_{naa} and Def_{nua} do not.*

Proof. Let $E, E' \subseteq A,\ F = (A, R) \in U_F$ such that $E \subseteq E' \subseteq \chi_\delta(E)$.
($\delta = Def_c$) Any $a \in \chi_c(E)$ is unattacked in $Free_c(E)$ and since $E' \subseteq Free_c(E)$ a is not attacked by E' in particular, so $\chi_c(E) \subseteq Free_c(E')$. Because $E \subseteq E'$ and Def_c is monotonic $Free_c(E') \subseteq Free_c(E)$ so every unattacked argument in $Free_c(E)$ is unattacked in $Free_c(E')$.
($\delta = Def_{cog}$) The same as the classic case, because all self-attackers are defeated by any E.
($\delta = Def_{lub}$) For an lub-admissible E the set $Free_{lub}(E) = E \cup gr^c(F^E)$ is conflictfree, so $E' \cap E'^+ = \emptyset$. Because $E' \setminus E \subseteq gr^c(F^E)$ it follows by the definition of c-grounded that $gr^c(F^E) \setminus (E' \setminus E) = gr^c(F^{E'})$. But then $\chi_{lub}(E) = \chi_{lub}(E')$, so the generalized Fundamental Lemma is satisfied.

For Def_{naa} resp. Def_{nua} Example 2.44 with $E = \emptyset$ and $E' = \{a, b\}$ suffices as a counterexample. □

For defeat operators satisfying the generalized Fundamental Lemma we can always construct a complete superset for any admissible extension by iterating the corresponding defense operator. Since the empty set is always admissible, this guarantees the existence of at least one complete extension for any AF $F \in U_F$, so the grounded semantics is well-defined.

Proposition 4.22. *Let $F = (A, R)$ be an AF. If δ satisfies the generalized Fundamental Lemma $co^\delta(F)$ is not empty and for any $E \in ad^\delta$ exists a δ-complete superset.*

Proof. If $E' \subseteq \chi_\delta(E)$ and $\chi_\delta(E) \subseteq \chi_\delta(E')$ then also $E' \subseteq \chi_\delta(E')$ so E' is δ-admissible. Since the generalized Fundamental Lemma holds for all admissible E this argument can be repeated for a proper superset of E' in $\chi_\delta(E')$ until a fixpoint $E' = \chi_\delta(E')$ is reached. For finite AFs the existence of such a fixpoint is guaranteed. As the empty set is always δ-admissible, at least one δ-complete extension must therefore exist. □

Note that the generalized Fundamental Lemma does not imply the single-status of the resp. grounded semantics. A counterexample for this is what we call reverse defeat.

Definition 4.23 (reverse defeat)**.** The *reverse* defeat operator is defined for any $F = (A, R) \in U_F$ and any $E \subseteq A$ by

$$Def_{rev}(F, E) \coloneqq \{a \in A \mid a \to E\}$$

The unique semantics family induced by reverse defeat is very similar to conflictfree semantics but has a nontrivial complete semantics and its grounded semantics is not single-status.

Proposition 4.24. *For all $F \in U_F$ it holds that $ad^{rev}(F) = ad^{cf}(F)$ and $pref^{rev}(F) = na(F)$, but there exist some AF such that $co^{rev}(F) \subset ad^{rev}(F)$ and in some cases $gr^{rev}(F) \neq \{\emptyset\}$ and rev-grounded extensions are not always unique.*

Proof. If E is not conflictfree it defeats the attacking arguments it contains, so $E \not\subseteq Free_{rev}(E)$ and thus not rev-admissible. If E is conflictfree, it defeats all outer attackers, so it is unattacked in $Free_{rev}(E)$. The rev-preferred extensions are thus the maximal conflictfree sets. For a rev-admissible extension that is not rev-complete consider Example 2.2. where the empty set is conflictfree but defeats no arguments and therefore defends a. So $\{a\}$ is rev-grounded. Since $\{b\}$ is conflictfree and defeats a, $\{b\}$ is another rev-grounded extension. □

The reverse defeat operator satisfies the generalized Fundamental Lemma because $E \subseteq E'$ implies $E^- \subseteq E'^-$ and thanks to the conflictfreeness of $\chi_{rev}(E)$ (Theorem 3.10) there are no conflicts among E' to be feared so defense is maintained. But Example 2.2 shows there can be multiple rev-grounded extensions. So unlike the Knaster-Tarski-Theorem the generalized Fundamental Lemma is not sufficient to guarantee the single-status of the grounded semantics although it guarantees the existence of grounded extensions. Another consequence from the generalized Fundamental Lemma is that preferred extensions are always complete because all admissible extensions have complete supersets.

Corollary 4.25. *Let δ be a defeat operator. If δ satisfies the generalized Fundamental Lemma then $pref^{\delta}(F) = \{E \in co^{\delta}(F) \mid E \ \subseteq -maximal\}$ for all $F \in U_F$.*

Proof. On the one hand a δ-complete extension is always δ-admissible, so it has a δ-preferred superset if it is not δ-preferred itself. On the other hand by Prop. 4.22 a δ-preferred E extension is δ-admissible, so it has a δ-complete superset. Since preferred extensions cannot have proper admissible supersets, E has to be δ-complete itself. □

The interesting part about this Corollary is that the generalized Fundamental Lemma is not necessary for it. Both Def_{naa} and Def_{nua} do not satisfy the lemma. Nonetheless Def_{naa} has only complete preferred extensions. In contrast there exist nua-preferred extensions which are not nua-complete like for Example 3.31, so completeness of preferred extensions does not hold for any defeat operator. This hints at the existence of a weaker condition for the equality in Cor. 4.25. It probably has to do with the modularization property but up to this point the true nature of such a condition remains unclear.

4.4 Singleton-additivity

Computational complexity is yet another topic which did not fit in the limited scope of this work. The complexity of a defeat-based semantics of course depends on the resp. defeat operator. Thus, without going into detail, the complexity of defeat operators makes a huge difference under the new defense notion. A convenient property in this regard is singleton-additivity.

Definition 4.26 (singleton-additivity)**.** Let $F = (A, R)$ be an AF, $op : 2^A \to 2^A$ an operator on the set of argument sets. op is *singleton-additive* iff for all $E \subseteq A$[6]

$$op(E) = \bigcup_{a \in E} op(\{a\})$$

Defeat operators satisfying this property can be defined on argument-level. The defeated arguments by a set E are exactly those defeated by its members. This is yet another property inherent to the classic concept *defense by attack*. Singleton-additivity also applies to cogent defeat, if the empty set is handled as an exception. If our goal is ignoring self-attackers on top of classic defeat, cogent semantics therefore can most likely not be topped efficiency-wise.

Proposition 4.27. *Def_c is singleton-additive, Def_{cog} is singleton-additive for nonempty sets only and Def_{naa}, Def_{lub} and Def_{nua} are not singleton-additive.*

Proof. $Def_c(E) = E^+ = \bigcup_{a \in E} \{a\}^+$ and for cogent semantics the self-attackers are defeated by any argument on its own so they are included in the union of all $Def_{cog}(\{a\})$. But since the epmty set also defeats all selfattackers, we have $Def_{cog}(\emptyset) \neq \emptyset = \bigcup_{a \in \emptyset} Def_{cog}(\{a\})$, so Def_{cog} is not singleton-additive for the empty set. □

For the other three semantics think of the following counterexample.

Counterexample 4.28. Example 2.44 continued. There $\{a, b\}$ no longer defeats b for any of these defeat operators, while $\{a\}$ alone does.

Singleton-additivity is not only relevant when it comes to complexity. On a more conceptual level it also demonstrates how much a defeat operator takes the structure of the remaining AF $A \setminus E$ into consideration when determining the defeated arguments. Recursive semantics are very meticulous in this regard, any little change of the reduct can have an impact on e.g. the arguments defeated under nua-defeat. As already demonstrated with the generalized Fundamental Lemma

[6] The range of applications is widened if the condition $E \neq \emptyset$ is added, from a practical point of view this should not cause any major problems

this makes adding arguments while maintaining admissibility very difficult. In case of naa-semantics modularization provides a certain degree of partitioning. The generalization of modularization for arbitrary defeat operators might be helpful to gain a better understanding of such structure-sensitive defeat operators in the future. In contrast singleton-additivity for defense operators leads us right back to our monotonicity problem of allowing no conflicting extensions.

Proposition 4.29. *Let $F = (A, R)$ be an AF and $op : 2^A \to 2^A$ an operator. If op is singleton-additive it is also monotonic.*

Proof. The union of sets is monotonic. □

Chapter 5

Defense-related principles for weak semantics

The original purpose of this thesis was to find principles suited for weak argumentation and to probe existing semantics for them. Chapter 5 is dedicated to this task. The focus of our new principles shall again be the defense behavior and the accepted arguments instead of characterizing the form of the semantics with properties like single-status, reinstatement or I-maximality. Relevant principles already in existence are directionality, modularization and unattack inclusion. Their focus is an intuitive interpretation of the AF-structure by a semantics. Directionality for instance grasps the intuition that the acceptability of an argument a is only influenced by arguments with a path to a.

5.1 Acceptance by default

Dondio and Longo take up the *in dubio pro reo* principle as their motivation to construct weak semantics based on Undecidedness Blocking over SCC-recursive algorithms in [Dondio and Longo, 2018]. Although other approaches to weak semantics do not directly mention *in dubio pro reo* they, too, are motivated by accepting arguments whenever no good reason for their rejection is available. The idea is something akin to stable semantics, but on the semantics level instead of single extensions. Arguments which are not creduously accepted by a semantics should at least be attacked by accepted arguments. We summarize this line of thinking as *default acceptance*. Default acceptance can be considered the central objective in the construction of weak semantics. In order to conduct a comparative study of weak semantics a formalization of this principle is therefore desirable.

As a first proposal, we translate the principle as is: An argument should be accepted if none of its attackers are acceptable.

Definition 5.1 (primitive acceptance by default)**.** A semantics ς satisfies the principle of *primitive acceptance by default* iff for any AF $F = (A, R)$ and any argument $a \in A$: If $b \notin_{ext} \varsigma(F)$ for all attackers $b \in \{a\}^-$ of a then $a \in_{ext} \varsigma(F)$.

The problem with this direct approach becomes obvious from the reverse statement.

Proposition 5.2 (acceptance by default over attacks)**.** *A semantics ς satisfies the principle of primitive acceptance by default iff for any AF $F = (A, R)$ and any argument $a \in A$: If $a \notin_{ext} \varsigma(F)$ then there exists an $E \in \varsigma(F)$ such that $E \to a$.*

Example 5.3. In Example 2.41 a is not accepted by, for instance, lub-semantics despite having only c as an attacker which is also not accepted.

In cases like Example 5.3 acceptance leads to a contradiction. An argument ends up being inacceptable not due to a legitimate attacker but due to the structure of its conflicts with other arguments. Dondio and Longo solve this dilemma by demanding that the inacceptance of an attacker b of a is only legit if it is not related to a and consequently proceed with SCC-recursive semantics.

We need not make limitations like this if the semantics can be defined over a defeat operator. Instead of working with direct attacks on a we argue it suffices to show that a is *defeated* by an accepted extension.

Definition 5.4 (acceptance by default over defeat)**.** A semantics ς satisfies the principle of *acceptance by default* iff for any AF $F = (A, R)$ and any argument $a \in A$: If $a \notin_{ext} \varsigma(F)$ then there exists an $E \in \varsigma(F)$ such that $a \in Def_\varsigma(E)$.

This generalization can be justified by the intuitive principle of rejecting arguments that one can successfully defend against. If and only if an accepted extension considers an argument defeated we have a good enough reason to reject that argument. All arguments not satisfying this condition should be accepted. Note that we do not ask for the reverse to be true, that is, we do not want all defeated arguments to be inacceptable.

We will now examine the four weak semantics introduced, as well as classic and conflictfree semantics for reference, on whether they satisfy defeat-based default acceptance. It makes little sense to ask this property of a grounded semantics. The other non-classic semantics in this work mostly satisfy default acceptance, except for cogent semantics. Since cogent semantics only rejects self-attackers and arguments directly attacked by the extension in question it cannot explain why the arguments in Example 5.3 are not acceptable. And of course classic semantics does not satisfy the principle for the same reason.

Proposition 5.5.

I ad^{naa}, co^{naa}, $pref^{naa}$, ad^{nua}, *all four lub- and all four cf-semantics satisfy the principle of acceptance by default.*

II gr^{naa} *and all four c- as well as all four cog-semantics do not.*

III $pref^{nua}$, co^{nua} *and* gr^{nua} *are an open question.*

Proof. For ad^{naa} and ad^{nua} every argument $a \notin \bigcup_{E \in \varsigma(F)} E$ is an element of $Def_{\varsigma}(\emptyset)$ and since $\emptyset$ is always admissible, a is defeated by an admissible extension. For naa-semantics the status of not being naa-admissible is hereditary in the reducts of naa-admissible extensions, so co^{naa} and $pref^{naa}$ satisfy the principle too. In more detail: First consider that all naa-admissible arguments are also naa-complete and naa-preferred, because naa-preferred extensions are the maximal naa-admissible extensions and are proven to be naa-complete. Now any argument that is not naa-admissible is either directly attacked by or not naa-admissible in the reduct of any naa-admissible extension (that includes naa-complete and -preferred extensions as special cases) because of the modularization property. Because if a non-admissible argument a would be naa-admissible in the reduct of an naa-admissible E, the join of E and the naa-admissible extension containing a in the reduct would be naa-admissible in the original AF. Contradiction.
For lub-semantics we know that the empty set defeats every argument that is not in $gr^c(F)$ and that $gr^c(F)$ is lub-admissible. The lub-grounded semantics is the classic grounded, but with a different defeat operator. If $gr^{lub}(F^{gr^c(F)})$ was not empty, $gr^c(F)$ would not be complete, so gr^{lub} satisfies the principle. Since the lub-grounded extension is complete, co^{lub} satisfies the principle too.
In case of conflictfree semantics we defeat every argument that is not a member of the extension in question, so $Def_{cf}(\emptyset) = A$, i.e. the empty set defeats all arguments. For naive semantics every argument which is not a self-attacker is contained in at least one naive extension and every self-attacker is defeated by any naive extension. □

Counterexample 5.6. For classic and cogent semantics consider Example 2.41. For gr^{naa} consider Example 2.22 as a counterexample, where the only naa-grounded extension $\emptyset$ does not defeat neither a nor b.

The difference defining default acceptance with defeat instead of attack makes can be seen in the lub-grounded semantics. It yields the same extensions as the classic grounded semantics but in contrast to it gr^{lub} satisfies default acceptance because it defeats more arguments. The tendency is the weaker the semantics the easier it is to satisfy default acceptance with cf-semantics implementing the principle as it is in Def_{cf}. The two recursive semantics are striking a tough balance between defeating and accepting arguments e.g. the naa-grounded semantics is skeptical enough not to satisfy default acceptance, while co^{naa} proves to be a proper completion of ad^{naa} in this regard, too. co^{nua} and gr^{nua} would satisfy the principle if such extensions always existed (which is an open question) because an nua-complete extension must include every argument it does not defeat. To see this consider an nua-admissible extension D in the reduct of an nua-complete extension E. If D is attacked by another nua-admissible extension in the reduct it is defeated, if not, it is unattacked in $Free_{nua}(E)$. The positive results for the weaker semantics in particular make default acceptance a principle for how weak a semantics implementing weak argumentation should be at least.

5.2 The separation property

Default acceptance demands that unaccepted arguments have no influence on the acceptance of arguments they attack. The *separation property* takes this on a structural level and demands a reverse directionality in case unaccepted arguments are the link between an unattacked set and the rest of an AF. In short a semantics satisfying the separation property should treat the rest of the AF as if those unaccepted attackers did not exist. What we mean by this will become clear with the following example.

Example 5.7. Example 2.41 continued. If a semantics ς does not accept the arguments a, b, c of the 3-cycle, d should be treated by ς the same way as an unattacked argument and e as if it was attacked by an unattacked argument.

Lub-semantics for instance do not treat the arguments d, e as if the 3-cycle did not exist and in Section 2.3.2 we describe this unintuitive behavior of unaccepted arguments influencing the acceptance of others. We now formalize this intuition by defining a weak and a strong version of immunity against attacks from unaccepted members of an unattacked argument set.

Definition 5.8 (SEP-I)**.** A semantics ς satisfies the *Separation Property I* if for any AF $F = (A, R)$ and any unattacked subset $F_1 \subseteq A$ with $\varsigma(F_1) = \{\emptyset\}$ it holds that $\varsigma(F \setminus F_1) = \varsigma(F)$.

Definition 5.9 (SEP-II)**.** A semantics ς satisfies the *Separation Property II* if for any $F = (A, R)$ and any unattacked subset $F_1 \subseteq A$ the following holds: If for every $a \in F_1$ with $a \rightarrow (F \setminus F_1)$ it holds that $a \notin_{ext} \varsigma(F_1)$, then

$$\varsigma(F \setminus F_1) = \{E \cap (A \setminus F_1) \mid E \in \varsigma(F)\}$$

Note that SEP-II implies SEP-I, because when all members of the unattacked set are unacceptable this includes the arguments attacking the rest of the AF. The parallels to directionality(see Def. 2.32) are easy to see and intended. For example, to make sure that the unacceptability of the arguments in F_1 is independent from the rest of the AF we apply the semantics in question directly on F_1. By doing this we can check if the semantics in question handles the setting of an unacceptable unattacked set (unacceptable attackers from this set resp.) as if it was directional even if it is not. Indeed there exist semantics which satisfy separation but not directionality, e.g. the naa-complete semantics. This is partly due to the directionality of ad^{naa} and partly due to modularization. Let us start with proving both separation properties hold for the naa-admissible semantics.

Proposition 5.10. *ad^{naa} satisfies both SEP-II and SEP-I.*

Proof. This will be proven for SEP-II by induction over the size of $F_2 := F \setminus F_1$, SEP-I is then included. The base case $F_2 = \emptyset$ is trivial.
Let $\#(F_2) = n+1$. The case $E = \emptyset$ is trivial too, so suppose $E \neq \emptyset$.Suppose first $E \in ad^{naa}(F_2)$.
Then F^E together with ${F_2}^E$ satisfy the induction hypothesis, since F_1 is not attacked by F_2. For any attacker a of E it now holds that either

$a \in F_1$, then $a \notin_{ext} ad^{naa}(F^E)$ because of the directionality of ad^{naa}

or $a \in F_2$, then $a \notin_{ext} ad^{naa}(F^E)$ iff $a \notin_{ext} ad^{naa}({F_2}^E)$ by the induction hypothesis

Therefore $E \in ad^{naa}(F)$.
Suppose now $E \in ad^{naa}(F)$. The case $E \cap F_2 = \emptyset$ is trivial, so suppose this is not the case. The precondition of SEP-II implies both that E does not attack F_2 and that every $a \in ad^{naa}({F_1}^E)$ does not attack F_2, because by Prop.2.53 all attackers of F_2 in F_1 are still not naa-admissible in ${F_1}^E$. So again F^E and ${F_2}^{E \cap F_2}$ satisfy the induction hypothesis. Now because of modularization $E \cap F_2$ is naa-admissible in $F^{E \cap F_1}$ and because of the induction hypothesis it is therefore also naa-admissible in F_2. □

It is no surprise that the naa-admissible semantics satisfies separation because an unattacked set remains unchanged in the reduct of extensions from the rest of the AF. The interdefinition of naa-defeat and naa-admissibility carries this result on to the naa-complete and naa-preferred semantics while the naa-grounded semantics is too sceptical to satisfy separation.

Corollary 5.11. *The naa-complete and naa-preferred semantics satisfy both SEP-II and SEP-I, the naa-grounded satisfies neither.*

Proof. Note first that the arguments credulously accepted under ad^{naa} are precisely the arguments accepted under $pref^{naa}$ and co^{naa} because naa-preferred extensions are $\subseteq$-maximal in ad^{naa} and always complete. For naa-preferred SEP-II now follows directly from Prop.5.10, because $\subseteq$-maximality is preserved under cuts.
Since ad^{naa} satisfies SEP-II, Def_{naa} satisfies it too, in the sense that
$Def_{naa}(E) \cap (F_2) = Def_{naa}(E \cap F_2)$ on F and on F_2 respectively for any naa-admissible $E \in ad^{naa}(F)$, and if the defeat operator is the same so is the defense operator. So if E is naa-complete on F then $E \cap F_2$ is naa-complete on F_2. Suppose now E is naa-complete on F_2. Then by modularization E can be combined with any naa-complete extension E' of F_1 and $E \cup E'$ will be naa-complete (see in detail in the proof of Cor.5.12).
For the naa-grounded semantics take Example 2.22 as a counterexample, the naa-grounded extension of the unattacked $F_1 = \{a, b\}$ is the empty set, but c is not naa-grounded in F even though it is unattacked in $A \setminus F_1$. □

The naa-grounded semantics would satisfy both properties if their condition was for the naa-admissible extensions of F_1 to be the empty set (contain no attackers of $F \setminus F_1$ resp.) instead. For defeat-based semantics this would be a reasonable weakening, however we would like the separation property to stay as general as possible, especially to make a comparison of naa-semantics with the SCC-recursive semantics from [Dondio and Longo, 2021] possible in the future. The above proof also demonstrates a limited directionality of the naa-complete semantics, namely:

Corollary 5.12. *Let $F = (A, R)$ be an AF, $F_1 \subseteq A$ unattacked and suppose no attacker $a \in F_1$ of F_2 is naa-admissible.*[1] *Then $co^{naa}(F_1) = co^{naa}(F) \cap F_1$.*

So as long as no naa-complete argument of the unattacked set attacks the rest of the AF co^{naa} is directional. The core of the proof for this is the modularization property. One could say the non-admissible arguments function as a barrier between the unattacked set and the rest of the AF and thus enable the combination

[1] Equivalently one can require a to not be naa-complete

of the complete extensions of the unattacked set with admissible extensions of the rest through modularization.

Proof. ($\supseteq$) can be proven with reasonable effort, use directionality of ad^{naa}. For the other direction suppose $E \in co^{naa}(F_1)$ and let $F_2 = F \setminus F_1$. F_2 has at least one naa-complete extension, because the empty set is naa-admissible and every naa-admissible extension has an naa-preferred superset which is always naa-complete, choose any such naa-complete extension E'. Because of Prop.2.53 and 5.10 E' is naa-admissible in F^E. Suppose now E' was not naa-complete in F^E. Then some $a \in \chi_{naa}(E') \setminus E'$ would exist. Because $F^{E^{E'}}$ satisfies SEP-II too and E' is naa-complete in F_2, $a \notin F_2$. But because of directionality a cannot be in F_1 either, or E would not be naa-complete. So E' is naa-complete in F^E and therefore $E \cup E'$ is naa-complete in F. □

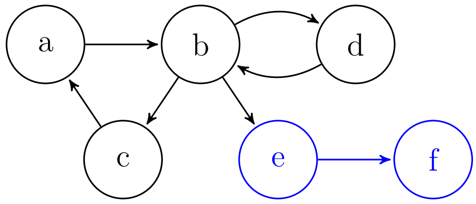
The separation property or the intuition behind it was part of our motivation for introducing nua-semantics. It turns out ad^{nua} satisfies only SEP-I. This proves the distinction between SEP-I and SEP-II is significant. As with default acceptance the nua-complete semantics has to remain as an open question because the existence of nua-complete extensions is not proven.

Proposition 5.13. *ad^{nua} and $pref^{nua}$ satisfy SEP-I but none of the nua-semantics satisfies SEP-II, gr^{nua} satisfies neither.*

Proof. Since F_1 is unattacked, for any $E \subseteq F_2 = A \setminus F_1$ we have $F_1 \subseteq F^E$ and F_1 is unattacked in this reduct. By directionality of ad^{nua} therefore $F_1 \cap ad^{nua}(F^E) = \emptyset$ so all arguments of F_1 are defeated by E, thus $E \in ad^{nua}(F_2)$ implies $E \in ad^{nua}(F)$. For the other direction by the precondition of SEP-I, if $E \in ad^{nua}$, then $E \subseteq F_2$ and F^E now satisfies SEP-I by induction over the size of F_2, so $Def_{nua}(E)$ on F is $F_1 \cup Def_{nua}(E)$ on F_2, so $E \in ad^{nua}(F_2)$.
For $pref^{nua}$ we can apply $\subseteq$-maximality like for $pref^{naa}$ and Example 2.22 serves again as a counterexample for gr^{nua}. □

Next, let us provide a counterexample for SEP-II. Just like lub-semantics, nua-semantics accept too much. The problem here though is not so much the propagation of undecidedness but the missing modularization or, more specifically the missing persistent non-admissibility resulting from it (Prop. 2.53). An argument which is not nua-admissible might become nua-admissible in some reduct and this interferes with the separation as the example demonstrates.

Counterexample 5.14. In the following AF $\{c, f\}$ is nua-admissible(and -complete, and -preferred), while $\{f\}$ is not in $F{\downarrow}_{\{e,f\}}$. The reason for this lies in $\{b\}$ being nua-admissible in $F^{\{c,f\}}$ and thus nua-defeating the attacker e of f.



SEP-I and SEP-II formalize concepts of non-interference which are not reached neither by too strong nor too weak semantics. C-semantics do not satisfy them because non-admissible attackers cannot be ignored, they accept too little. Neither do Lub-semantics, because they also ignore attackers attacked by non-admissible arguments and end up accepting too much. Therefore the separation property may become a cornerstone for developing a balanced weak semantics.

Proposition 5.15. *ad^*, co^* gr^* $pref^*$ do not satisfy neither of the separation properties for $* \in \{c, cog, lub\}$*

Proof. Example 2.41 serves as a counterexample, for c- and cog-semantics do not accept d and as mentioned before lub-semantics accepts e, while all three semantics reject the arguments of the unattacked 3-cycle a, b, c and would accept d and reject e in $F{\downarrow}_{\{d,e\}}$. □

5.3 Minimal acceptance and minimal rejection

If the separation property is about the right balance between ignoring and accepting arguments, this section is about extremes. We discuss which arguments a weak or more generally speaking a defense-oriented semantics has to accept no matter what and which should always be rejected. A first approach towards this can be found in [Baumann et al., 2020a]. There the authors propose a property called *unattack inclusion* in order to characterize the c-grounded semantics in terms of modularization.

Definition 5.16 (unattack inclusion)**.** A semantics ς satisfies *unattack inclusion* iff a unattacked implies $a \in_{ext} \varsigma(F)$ for all $a \in A$, $F = (A, R) \in U_F$.
ς satisfies *skeptic unattack inclusion* iff a unattacked implies $a \in E$ for all $E \in \varsigma(F)$.

Skeptic unattack inclusion was added by us as a form of minimal completeness (the name strict is already used differently in [Baumann et al., 2020a]). It is hard to argue against the necessity for an argumentation semantics to accept unattacked arguments. An unattacked argument is defended by default since it has no attackers. Therefore it should at least be credulously accepted and any semantics claiming to be complete defense-wise should include unattacked arguments in all of its extensions because they are defended under any circumstances. We dare say this intuition is shared by the majority of argumentation scientists. The question now is how to guarantee unattack inclusion. Incase of defeat-based semantics a simple criterion is sufficient for unattack inclusion (although not necessary) for admissible semantics and a more specific criterion for sceptic unattack inclusion suffices for defeat-based complete semantics.

Proposition 5.17. *Let δ be a defeat operator.*

I. If δ is selfdefeatfree, ad^{δ} satisfies unattack inclusion.

II. ad^{δ} never satisfies skeptic unattack inclusion.

III. If $\delta(E) \cap \{a \in A \mid a \text{ unattacked}\} = \emptyset$ for all $E \subseteq A,\ F = (A, R) \in U_F$, then co^{δ} satisfies skeptic unattack inclusion.

Proof. Let $F = (A, R)$, $a \in A$. If a is unattacked than $\{a\}$ is conflictfree and therefore by Def. 4.4 $\{a\} \subseteq Free_{\delta}(\{a\})$ for any selfdefeatfree δ. Since a is unattacked, it follows $\{a\} \subseteq \chi_{\delta}(\{a\})$, so $a \in_{ext} ad^{\delta}$. The general admissiblity of the empty set(3.9) is enough to disprove skeptic unattack inclusion. The condition of the second statement ensures a unattacked implies $a \in Free_{\delta}(E)$ and unattacked there, which in turn implies $a \in \chi_{\delta}(E) = E$ for all $E \in co^{\delta}$. □

Basically we only need to exclude unattacked arguments from defeat for the relevant extensions. For simple unattack inclusion a selfdefeatfree operator is enough, so it comes as no surprise the four weak admissible semantics investigated in this work all satisfy unattack inclusion.[2]

Corollary 5.18.

I. $ad^{naa}, ad^{lub}, ad^{nua}, ad^{cog}$ satisfy unattack inclusion but not skeptic unattack inclusion.

II. $co^{naa}, gr^{naa}, pref^{naa}, co^{lub}, gr^{lub}, pref^{lub}, co^{cog}, gr^{cog}, pref^{cog}$ satisfy skeptic unattack inclusion.

Proof. (I) follows from Prop. 5.17 and Prop. 4.5.
(II) Since all semantics here are either complete or subsets of the resp. complete

[2]for the naa-semantics family this was already proven in [Baumann et al., 2020a]

semantics, it is sufficient that the defeat operators in question satisfy Prop. 5.17 which can be seen from their resp. definitions. □

Apart from the nua-complete and nua-grounded with their open existential issues as well as the nua-preferred semantics with the missing completeness, the other semantics with fixpoint extensions all satisfy skeptic unattack inclusion. We have reasons to believe the same holds for the three nua-semantics but the proof for this has to be left open at this point.

If we accept unattacked arguments why should we reject the arguments c-defended by them? Whether all c-accepted arguments have to be accepted under a semantics with a different defense concept might be open to discussion, but c-grounded arguments are something one cannot reason against in a real discussion. We therefore argue they should receive the same treatment as unattacked arguments, at least from semantics which aim to *weaken* c-defense.

Definition 5.19 (c-grounded inclusion)**.** A semantics ς satisfies *c-grounded inclusion* iff $a \in_{ext} gr^c(F)$ implies $a \in_{ext} \varsigma(F)$ for all $a \in A$, $F = (A, R) \in U_F$.
ς satisfies *skeptic c-grounded inclusion* iff $a \in_{ext} gr^c(F)$ implies $a \in E$ for all $E \in \varsigma(F)$.

The four weak semantics in question here have no problem with this additional restriction, the proof for the three nua-semantics has to wait, again. We also expect the SCC-recursive semantics from [Dondio and Longo, 2021] to satisfy c-grounded inclusion.

Proposition 5.20.

I. $ad^{naa}, ad^{lub}, ad^{nua}, ad^{cog}$ *satisfy c-grounded inclusion but not skeptic c-grounded inclusion.*

II. $co^{naa}, gr^{naa}, pref^{naa}, co^{lub}, gr^{lub}, pref^{lub}, co^{cog}, gr^{cog}, pref^{cog}$ *satisfy skeptic c-grounded inclusion.*

Proof. (I) follows from Prop. 4.2 as the c-grounded extension is c-admissible.
(II) We know all these semantics satisfy skeptic unattack inclusion and that the defeat operators in question all defeat at least E^+ for any $E \in \varsigma(F)$, $F = (A, R)$ an AF, ς one of the above semantics. So unless they are defeated all arguments which are c-defended solely by unattacked arguments are unattacked in $Free_\delta(E)$ and therefore included in E if E is δ-complete. This can be repeated for every iteration step of the c-grounded semantics. Suppose $a \in_{ext} gr^c(F)$ such that E contains c-grounded attackers for each attacker of a. Then E cannot attack a

because E is conflictfree, so a in F^E and unattacked there. But then $a \in \chi_\delta(E)$ for the defeat operators of all the above semantics so $a \in E$ because all are complete semantics. □

As important as it is to include indisputable arguments is it not to include indisputably defeated arguments. The question in which cases an argument has to be rejected is not as straightforward as acceptance. We just made a strong case that c-grounded arguments are as strongly defended as unattacked arguments. By doing this we already commit ourselves to reject arguments attacked by unattacked or c-grounded arguments and we deem this reasonable. A defense against unattacked attackers in particular we find rather absurd, it undermines the integrity of the attack relation as our basis for reasoning about a given AF. And for the same reason for which we argued that c-grounded inclusion is a better option of minimal acceptance we now propose c-grounded rejection as a minimal rejection criterion for weak semantics (and unattack rejection as a weaker alternative).

Definition 5.21 (rejection)**.** A semantics ς satisfies *c-grounded rejection(unattack rejection)* iff for every $F = (A, R) \in U_F$ and every $a \in A$ such that a is attacked by some $b \in_{ext} gr^c(F)$ (b unattacked) it holds that $a \notin_{ext} \varsigma(F)$.

Note that this is the first property of Chapter 5 which is not satisfied by conflictfree semantics (besides skeptic unattack/c-grounded inclusion). Default Acceptance, Separation Property, unattack inclusion - conflictfree semantics do exceedingly well as weak semantics. Their main fault lies exactly within rejection, they simply accept too many arguments. The four weak semantics between cf- and c-semantics on the other hand respect c-grounded rejection just as rigorously as they respect c-grounded inclusion.

Proposition 5.22. *All semantics based on the defeat operators Def_c, Def_{cog}, Def_{lub}, Def_{naa} and Def_{nua} satisfy c-grounded (and unattack) rejection.*

Proof. It is enough to show this for the resp. admissible semantics. Note first that none of the defeat operators defeat unattacked arguments(for nua-semantics this follows from any unattacked argument being an unattacked nua-admissible extension by itself in any reduct), so unattack rejection is guaranteed. For c-grounded rejection we use the strong admissibility of the c-grounded semantics. Let $F = (A, R)$, $E \in \varsigma_\delta(F)$ where δ is one of the defeat operators above. Suppose E is attacked by some $a \in_{ext} gr^c(F)$. Since such an a is no self-attacker, for E to be c- or cog-admissible E has to attack a back. Strong admissiblity now implies the existenc of some $b \in_{ext} gr^c(F), b \neq a$ that defends a by attacking E, so E has to defeat b too and then by strong admissibility some $c \in_{ext} gr^c(F) \setminus \{a, b\}$ exists that attacks E and so on until E can no longer attack, ultimately till an unattacked $u \in_{ext} gr^c(F)$ is reached. So E cannot be c- or cog-admissible if it is attacked by a c-grounded argument. In case $E \in ad^{lub}/ad^{naa}/ad^{nua}(F)$ if E is attacked by some $a \in_{ext} gr^c(F)$, because of strong admissibility some c-grounded attacker b of E will always remain in the reduct unless E is not conflictfree but then it is rejected. Strong admissibility guarantees the existence of such a c-grounded attacker b that is also c-grounded in the reduct. For naa-semantics and lub-semantics this suffices to show the proposition. For nua-semantics we need an induction over the size of F. The c-grounded extension of the reduct F^E of any nonempty $E \in ad^{nua}(F)$ cannot be attacked by any nua-admissible extension in the reduct, because c-grounded rejection is satisfied in the reduct according to the induction hypothesis and c-admissibility states that the c-grounded extension of the reduct attacks every argument it is attacked by. So E cannot defeat a c-grounded attacker that is also c-grounded in the reduct, so E is not nua-admissible. Contradiction. □

5.4 On the influence of defeated arguments

Our defense notion introduced in Chapter 3 relies heavily on the (correct) choice of the defeat operator. Classic admissibility is a simple case in this regard, since defeat coincides with attack. Weaker semantics have to make complex situational rulings about which extensions are to be accepted and which not. Defense is not always considered separately but together with other conditions extensions have to satisfy (e.g. the original definition of lub-complete semantics). In these cases the following criterion can be used to identify the defeated arguments under a certain semantics as well as to validate potential/suggested defeat operators.

Definition 5.23 (defeat criterion)**.** Let $F = (A, R)$ be an AF and let ς be a semantics. An argument a is defeated by an extension $E \in \varsigma(F)$ w.r.t. ς iff $E \in \varsigma(F')$ for any $F' = (A, R \cup add)$ with an additional set of attacks $add \subseteq \{(a, e) \mid e \in E\}$ from a to E.[3]

[3] We presuppose that a is not part of the extension itself because otherwise this condition would always lead to a violation of conflictfreeness. Since the extensions of a defeat operator based semantics never defeat their own arguments(3.10) and since it would be strange for a semantics to allow defeat within an extension, we believe this restriction is reasonable.

The intuition behind this defeat criterion is that adding attacks from defeated arguments should no longer be able to cause any damage to the extension they are defeated by. This can be taken one step further by requesting that no attacks by a defeated argument to whatever other argument cause a problem.

Definition 5.24 (irrelevance criterion)**.** Let $F = (A, R)$ be an AF and let ς be a semantics. An argument a is rendered irrelevant by an extension $E \in \varsigma(F)$ w.r.t. ς iff $E \in \varsigma(F')$ for any $F' = (A, R \cup add)$ with an additional set of attacks $add \subseteq \{(a, b) \mid b \in A\}$.

While reduct-based semantics have no problem with the former, only cf-, c- and cog-admissible semantics satisfy the later criterion. Again cogent semantics can shine as a very stable weakening of classic defense.

Proposition 5.25. *Let $F = (A, R)$ be an AF and let $\delta \in \{c, cf, naa, lub, nua, cog\}$. Then for any $E \in \varsigma_\delta(F)$, where $\varsigma \in \{ad, co\}$ and any $a \in A$ it holds that $a \in \delta(E)$ iff a satisfies the defeat criterion.*
If $\delta \in \{c, cf, cog\}$ the irrelevance criterion can be used instead.

Proof. ($\Leftarrow$) Suppose $a \notin \delta(F, E)$ for any of these defeat operators, then $a \in Free_\delta(E)$ and adding a single attack from a to E in F' would result in E being no longer unattacked in $Free_\delta(E)$ and thus no longer δ-admissible (-complete, etc.). Because adding an attack from a to E would not change the status of a in the reduct the only cases in which no F' exists such that $a \notin \delta(F', E)$ still holds are those where a is the only member of E and the only way to add an attack is for a to attack itself, but then E is no longer conflictfree and therefore also no longer δ-admissible. Adding more options for F' with the irrelevance criterion does not change anything for this direction of the proof.
($\Rightarrow$) Suppose now $a \in \delta(F, E)$. Note that this implies $a \notin E$ by Theorem 3.10. Then as long as $\delta(F, E) = \delta(F', E)$ the δ-admissibility does not change, no matter how many attacks from a are added, because E stays unattacked in $Free_\delta$. So the question is whether a and the other defeated arguments stay defeated in any F'. Starting with the semantics satifying irrelevance, Def_{cf} asks only whether $a \in E$ or not and if $a \in Def_{cf}(E)$ then $a \notin E$(same holds for any other defeated argument b) and no amount of added attacks in F' can change that. The same holds for Def_c if $a \in E^+$, adding attacks will not help, and for Def_{cog} because adding attacks from a neither change E^+ nor the set of self-attackers (except that a may become a self-attacker, but that does not change anything because then a was in E^+ to begin with).

The cf-complete are cf-admissible extensions and the empty set is always the cf-grounded extension. As for naive semantics, adding attacks outside of E can only turn a conflictfree extension in a naive extension but not the other way around. For c-complete the set of c-defended arguments depends only on E^+ and since E is c-admissible we can conclude from $a \in Def_c(E)$ that a is not c-defended by E, so attacks from a will not change the completeness of E. The same applies to co^{cog}. For the other three semantics adding attacks wildly can cause a change in the reduct F'^E and therefore in $\delta(F', E)$. Adding attacks on E, however, does not change the reduct at all, whether $a \in E^+$ or $a \in F^E$. And if the reduct does not change, $\delta(F, E) = \delta(F', E)$ for any $\delta \in \{naa, lub, nua\}$ (and thus also E being or not being a fixpoint does not change.) □

We leave the respective grounded and preferred semantics as future work. The two criteria demonstrate how helpful it is to distinguish between arguments that are no threat to an extension and arguments that are actually defeated by it when studying the behavior of (weak) semantics. In particular if one wants a concept of not only admissibility but also defense this difference is of the utmost importance. After reading the proof of Prop. 5.25 one might wonder if it is even possible to not satisfy the defeat criterion with a defeat-based semantics. The answer is yes, an intuitive example for this is the reverse-defeat semantics family introduced in Def. 4.23.
An example of a not (yet) defeat-based semantics that violates the defeat criterion is the c-grounded semantics when it is based on strong admissibility instead of classic defense.[4] SCC-semantics in general seem to be vulnerable in this regard because their evaluation of an AF relies on the structural property that attacks from one SCC to the other are unidirectional.

5.5 Duplication

Weak semantics take the structure of an AF into account when determining defeat and acceptance. It therefore becomes nontrivial to check whether certain structures are handled consistently by them. Directionality or the separation property are examples of structural properties we would intuitively expect from a (weak) semantics. The property we introduce now examines how well-defined a semantics is. The strength of the underlying defeat, if it exists, does not matter. We want to investigate in this section if a semantics accepts arguments *equally* under equal circumstances.

[4] see Def. 13 and 14 in [Baroni et al., 2011]

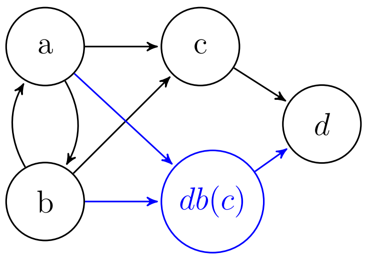
In order to do this we introduce the notion of an *argument double*. A double of an argument a has the same attack relations to other arguments as a. The result is an AF with two (or more if wished) arguments in the exact same positions.

Definition 5.26 (argument double)**.** Let $F = (A, R)$ be an AF, $a \in A$. A *double* of a is an argument $db(a)$ such that for all $b \in A$, $b \neq a$:

1. $b \to db(a)$ iff $b \to a$
2. $db(a) \to b$ iff $a \to b$
3. $db(a)$ attacks itself iff a attacks itself

We define $Db(F, a)$ to be the AF resulting from adding a double of a to F.

Example 5.27. The AF $Db(F, c)$ resulting from adding a double of c and the according attacks to Example 2.22 is



We now naturally would expect an argument and its double to defeat and defend the same arguments, to belong to the same or at least symmetrical extensions and so on. Most importantly the rest of the AF should not be evaluated differently no matter how many doubles of an argument are added because argumentation is about the relations between arguments not their numbers. For example with c-grounded semantics it does not matter how many arguments you attack with, if all those arguments are defeated by the same unattacked argument they carry no weight. The behavior we would expect from a well-defined semantics confronted with doubles can be summed up as follows.

Definition 5.28. A semantics ς is *stable under duplication* iff

$$\varsigma(Db(F, a)) = \{E \mid E \in \varsigma(F), a \notin E\} \cup \{E \cup \{db(a)\} \mid E \in \varsigma(F), a \in E\}.$$

holds for every $F = (A, R) \in U_F$, $a \in A$.

We limit ourselves to the case where a and $db(a)$ have to be in the same extensions and leave the investigation of symmetric extensions for later work. With the definition as it is we can only expect complete semantics and/or their subsets to satisfy it. An adequate criterion for the defeat operator of such a complete semantics is given below.

Proposition 5.29. *Let δ be a defeat operator. co^δ is stable under duplication iff*

$$\delta(Db(F,a),E) = \begin{cases} \delta(F,E) & a \notin \delta(F,E) \\ \delta(F,E) \cup \{db(a)\} & a \in \delta(F,E) \end{cases}$$

for all $E \subseteq A,\ F = (A,R) \in U_F$.

Proof. If $db(a)$ is defeated iff a is defeated, then the unattacked set $\chi_\delta(E)$ in $Free_\delta(E)$ does not change for $Db(F,a)$ except that $db(a)$ might be included. Since $db(a)$ is attacked by the same arguments as a it is unattacked in $Free_\delta$ iff a is, so a δ-complete extension has to contain it if it contains a. □

As one would expect classic and cogent semantics have no problem with these conditions. The three naa-semantics make the cut, too, because a double is only defeated if the original is defeated under naa-defeat. This does not hold true for co^{lub} or co^{nua} so both of them are not stable under duplication.

Corollary 5.30. *$co^\delta, pref^\delta$ and gr^δ are stable under duplication for $\delta \in \{c, cog, naa\}$, co^{lub} and co^{nua} are not.*

Proof. Let $F = (A,R)$ and $a \in A$. Def_{cog} satisfies Prop. 5.29, because $db(a)$ is attacked by the same arguments as a and is a self-attacker if a is, so it is defeated (resp. defended) by some E if a is. This proof suffices for Def_c too.
For Def_{naa} note that, since $db(a)$ and a attack the same arguments, $db(a)$ does not attack any argument in $Db(F,a)^{\{a\}}$ and the same holds for any E containing a but not $db(a)$. So for such an E we have $Db(F,a)^{E^{\{db(a)\}}} = F^E$ and E is naa-admissible iff $\{db(a)\}$ is naa-admissible in $Db(F,a)^E$. So while Def_{naa} does not satisfy the condition of Prop.5.29 (since $a \in E$ is not defeated but $db(a)$ is if both are attacked by an naa-admissible attacker of the reduct) the result in this case is nonetheless that $db(a)$ is unattacked in $Free_{naa}(E)$ if E is and therefore $db(a)$ is included in any naa-complete E containing a. For $a \notin E$ we have already demonstrated with the previous case that $db(a)$ is naa-admissible in $Db(F,a)^E$ iff a is not naa-admissible in F^E. Since both attack the same arguments in the reduct, E is a fixpoint in $Db(F,a)$ if it is in F.
Since minimality and maximality do not change by adding an argument to all complete extensions containing a, the above argumentation applies to the resp. preferred and grounded semantics too. □

Example 5.27 can serve as a counterexample for co^{nua} and co^{lub}.

Counterexample 5.31. In Example 5.27 $E = \{c\}$ is a da/lub-complete extension that contains c but not $db(c)$ because $db(c)$ is da/lub-defeated by E.

The problem with these two semantics is that their extensions defeat arguments they could also take in. This can be solved by requesting maximality.

Proposition 5.32. *gr^{lub}, $pref^{lub}$ and $pref^{nua}$ are stable under duplication.*

Proof. In the proof for co^{naa} we have shown that adding $db(a)$ to E does not impact the defeat status of the other arguments in the reduct, so adding $db(a)$ to an lub/nua-complete extension results in an lub/nua-complete extension which is preferred to the original. On the other hand any E that cannot take a in, cannot take $db(a)$ in also. For gr^{lub} the proposition follows from $gr^{lub} = gr^{c}$.[5] □

As always, an nua-semantics remains an open question, in this case gr^{nua}. Stability under duplication suits semantics with little structural dependency in their defeat, of course. The real challenge is to define a semantics as weak as possible under this condition. Naa-semantics satisfy it, the SCC-recursive semantics of [Dondio and Longo, 2021] are promising candidates. For future research this property could be formulated in terms of symmetric extensions to examine admissible semantics. An expansion of the concept to duplicating sets of arguments instead of arguments may lead to even stricter restrictions on the operation of semantics or their operators on structures and has the potential to deepen our understanding of recursive defense concepts.

[5] see Section 5.2 of [Dondio and Longo, 2021]

Chapter 6

ASPIC and the rationality of weak semantics

A lot of the discussed topics related to weak semantics circles around the rationality of ignoring attackers under certain circumstances. As mentioned at the beginning of Section 2.3 the fortitude of Dung-style argumentation lies in its close resemblance to the reasoning process of a human confronted with e.g. conflicting information. The role of argumentation semantics is to model the evaluation of these conflicts as closely as possible. The weak semantics introduced in this paper have a case in point here by yielding results closer to intuitive reasoning than c-semantics. The question is if these semantics can uphold the high standard of soundness one expects from an argumentation system for a real world application. For example, is a knowledge base derived with weak semantics consistent? When it comes to testing semantics for rationality the rationality postulates introduced in [Caminada and Amgoud, 2007] for the ASPIC Framework have received lots of positive attention.

The basic idea of the ASPIC Framework is to model formulas of a logical language as arguments and represent inconsistencies between them through the attack relation. Applying an argumentation semantics on the resulting AF shall then yield a set of acceptable conclusions from the given formulas. In principle, the modeling process with ASPIC follows three steps:

Input: A set of strict and a set of weak inference rules as well as a set of premises

1. Construct the AF from the given rule sets
2. Apply an argumentation semantics
3. Generate the set of acceptable conclusions from the acceptable arguments

Output: A set of conclusions

In order for these conclusions to be of any use we need them to satisfy certain criteria, e.g. consistency. The rationality postulates by [Caminada and Amgoud, 2007] propose three such criteria which are by now widely recognized. This chapter investigates ways in which the weak semantics introduced in this work may satisfy those postulates for ASPIC Frameworks and asks the question how they can contribute to the formalism. In order to do this Section 6.1 first gives a formal description of the ASPIC modeling process following [Modgil and Prakken, 2014]. Section 6.2 is dedicated to the rationality postulates from [Caminada and Amgoud, 2007], how they were realized for c-semantics with ASPIC$^+$ there and why this approach fails for weak semantics. Section 6.3 then introduces a relatively new ASPIC variation called *Deductive ASPIC*$^-$, which was proposed in [Cramer and Bhadra, 2020], and concludes this chapter by demonstrating how some of the weak semantics can satisfy the postulates under this formalism and why some of the others cannot.

6.1 ASPIC$^-$

We start with some specifications a logical language we want to model with ASPIC has to satisfy.

Definition 6.1 (logic prerequisites)**.** Let $\mathcal{L}$ be a fixed (defeasible) logic with L the set of all well-formed formulas under $\mathcal{L}$ and let L be closed under some (monadic) negation $\neg$. An instance of a strict inference rule is denoted by $p_1, p_2, ..., p_n \to q$ with $p_i, q \in L$ for all $i \in (1, ..., n)$, $n \in \mathbb{N}$,[1] an instance of a defeasible inference rule by $p_1, .., p_n \Rightarrow q$. We use $p = q$ to say that p and q represent the same well-formed formula from L (syntactic equivalence).

ASPIC is intended for defeasible reasoning(there is no need for argumentation in case of purely strict reasoning). The difference between defeasible inference rules and strict ones is crucial for the definition of attacks between conflicting formulae. A conclusion reached by applying defeasible rules may be doubted, one we arrived at with strict rule applications only is indisputable. At most, one could question the premises in the second case. For the formalism introduced here we refrain from this and consider a given set of premises set in stone. The premises and the formulae we derive from them are obvious candidates for arguments. In order to define attacks between them we need our logic $\mathcal{L}$ to have some sort of negation, a syntactic concept for conflicting formulae. For an easier and clearer handling of these conflicts, a symmetric notation for negation is used.

[1] $(1, ..., n) := \{i \in \mathbb{N} \,|1 \leq i \leq n\}$ is the (ordered) set of the natural numbers from 1 to n

Definition 6.2 (symmetric negation)**.** For $p, q \in L$ we define the symmetric negation function $- : L \to L$ by $-p \coloneqq q$ and $-q \coloneqq p$ iff $p = \neg q$ or $q = \neg p$.

We will now describe how to translate given formulae and inference rules of a logic $\mathcal{L}$ into an AF with ASPIC$^-$. We start with the arguments. In ASPIC arguments have an inner structure, they do not only contain a formulae derived from the given premises, but all steps of the inference process leading to said conclusion. Arguments are constructed inductively, we start with the premises and then apply the given strict and defeasible rules to successively generate all formulae resulting from our premises.

Definition 6.3 (argument construction)**.** Let $P \subseteq L$ be a set of premises, S a set of (instantiated) strict rules and D a set of (instantiated) defeasible rules. Then for (P, S, D) the set of arguments A and the function $Conc : A \to L$ which returns for each argument its conclusion are inductively constructed by:

For $p \in P$, $x_p = [p]$ is an argument with conclusion $Conc(x_p) = p$.

Let $s \in S$ with $s = p_1, p_2, ..., p_n \to c$. If there are arguments $x_1, x_2, ... \in A$ such that $Conc(x_i) = p_i$, then $x_s = [x_1, x_2, ..., x_n \to c]$ is an argument with $Conc(x_s) = c$.

Let $d \in D$ with $d = p_1, p_2, ..., p_n \Rightarrow c$. If there are arguments $x_1, x_2, ... \in A$ such that $Conc(x_i) = p_i$, then $x_d = [x_1, x_2, ..., x_n \Rightarrow c]$ is an argument with $Conc(x_d) = c$.

We say p, s or d is the *top-rule* of the resp. argument x_p, x_s or x_d.

Note that, depending on S and D, this construction method may lead to an infinite set of arguments.[2] We will limit ourselves to finite ASPIC frameworks since some of our weak semantics are well-defined on finite AFs only. For every argument $x \in A$ constructed the following properties are defined:

Definition 6.4 (argument structure)**.** Let A be a set of arguments constructed from a tuple (P, S, D). The functions Sub, $StrictRules$ and $DefRules$ are defined by

$Sub : A \to 2^A$ with $Sub(x_p) = \{x_p\}$ for any premise argument $x_p = [p]$,
$Sub(x_s) = x_s \cup \bigcup_{1 \leq i \leq n} Sub(x_i)$ for any argument $x_s = [x_1, ..., x_n \to c]$
$Sub(x_d) = x_d \cup \bigcup_{1 \leq i \leq n} Sub(x_i)$ for any argument $x_d = [x_1, ..., x_n \Rightarrow c]$

[2]An input (P, S, D) with $P = \{q\}, S = \{s_1 = q \to p\}, D = \{d_1 = p \Rightarrow q\}$ has such an effect since the rules s_1 and d_1 can be applied on each others conclusions in an infinite loop

$$\begin{aligned}
&StrictRules : A \to 2^S \text{ with } StrictRules(x_p) = \emptyset \text{ for } x_p = [p] \\
&\quad StrictRules(x_s) = s \cup \bigcup_i StrictRules(x_i) \text{ for } x_s = [[x_1, ..., x_n \to c] \\
&\quad StrictRules(x_d) = \bigcup_i StrictRules(x_i) \text{ for } x_d = [x_1, ..., x_n \Rightarrow c]
\end{aligned}$$

$$\begin{aligned}
&DefRules : A \to 2^D \text{ with } DefRules(x_p) = \emptyset \text{ for } x_p = [p] \\
&\quad DefRules(x_s) = \bigcup_i DefRules(x_i) \text{ for } x_s = [[x_1, ..., x_n \to c] \\
&\quad DefRules(x_d) = d \cup \bigcup_i DefRules(x_i) \text{ for } x_d = [[x_1, ..., x_n \Rightarrow c]
\end{aligned}$$

We now provide a simple example of an argument construction under ASPIC$^-$.

Example 6.5. Let $P = \{q\}$, $S = \{p \to \neg p\}$ and $D = \{q \Rightarrow p, q \Rightarrow t\}$. Then the tuple (P, S, D) yields the arguments

$$[q], [[q] \Rightarrow t], [[q] \Rightarrow p], [[[q] \Rightarrow p] \to \neg p]$$

E.g. for argument $x = [[[q] \Rightarrow p] \to \neg p]$ the top-rule is $p \to \neg p$, its subarguments are $Sub(x) = \{[q], [[q] \Rightarrow p], [[[q] \Rightarrow p] \to \neg p]\}$ and the strict/defeasible rules are $StrictRules(x) = \{p \to \neg p\}$ and $DefRules(x) = \{q \Rightarrow p\}$ respectively.

In order to define the attack relation on such an argument set, the following distinction is needed:

Definition 6.6. (defeasible argument) An argument x is said to be *strict* iff $DefRules(x) = \emptyset$, otherwise it is defeasible.

This allows us to restrict incoming attacks to the set of defeasible arguments, since an argument consisting only of strict rules shall not be doubted. An attack from an argument x to a defeasible argument y may now take one of two forms - Rebut or Undercut.[3] Informally x rebuts y if the conclusion of any subargument of x conflicts with the conclusion of any subargument of y. Undercut on the other hand is defined over a naming function.

Definition 6.7. (naming function) Let D be a set of instantiated defeasible rules. A partial naming function $n : D \to L$ assigns to some defeasible rules $d \in D$ a formula $p \in L$ as a name. The notation $q :\ p_1, ..., p_n \Rightarrow c$ is used instead of $q = n(d)$ for $d = (p_1, ..., p_n \Rightarrow c) \in D$.

The idea is that a defeasible rules d is only applicable as long as the corresponding formula $n(d)$ has not been disproven. By reaching the conclusion $\neg n(d)$ in a subargument, x undercuts the application of the defeasible rule d in argument y. We can now formulate the definition of attacks in ASPIC.

[3]We will exclude undermining attacks from the scope of this work, for details see [Modgil and Prakken, 2014]

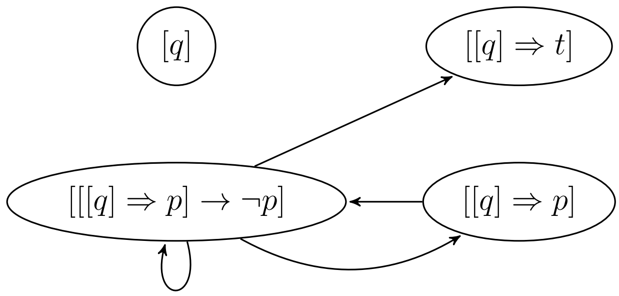
Definition 6.8. (attack relation) Let $(P, S, D(n))$ be a tuple of premises, strict rules, defeasible rules and a naming function and let A be the argument set constructed from them. For any $x, y \in A$ we say x attacks y, $(x, y) \in R$ iff y is a defeasible argument and one of the following conditions applies:

$Conc(x') = -Conc(y')$ for some $x' \in Sub(x)$, $y' \in Sub(y)$ (Rebut)

$Conc(x') = -n(d)$ for some $x' \in Sub(x)$ and some $d \in DefRules(y)$ (Undercut)[4]

Let us generate the attack relation for our previous example with this definition.

Example 6.9. Let (P, S, D) be the same as in Example 6.5 and let a partial naming function n be added to D: $D(n) = \{q \Rightarrow p, p : \ q \Rightarrow t\}$ The resulting ASPIC$^-$ framework $F = (A, R)$ is then



The idea of weak semantics to enable an extension to defend itself against for example self-attackers can become relevant for ASPIC frameworks too. In the above example a self-attacker undercuts an otherwise acceptable defeasible argument ruling it unacceptable. We might want to avoid this, since the conclusion $\neg p$ of the argument attacking the name $p = n([[q] \Rightarrow t])$ is not acceptable, so one could argue the defeasible rule $q \Rightarrow t$ should not be rejected yet. The usefulness of weak semantics in ASPIC lies exactly in those cases of unreasonable undercut. In an AF with only rebut attacks on the other hand an argument can counterattack such an unreasonable attacker, thus c-defending itself.[5].

[4]In [Caminada and Amgoud, 2007] the attack relation is introduced by defining attacks first and then defeat in terms of successful attack as the actual attack relation. Since the term defeat is already used in the context of defeat operators in this work, successful attacks are introduced directly as attacks here.

[5]It cannot defend itself in all cases under ASPIC$^+$, but for reasons we will explain later, we might not want it to defend itself in those cases anyway

6.2 The rationality postulates

In [Caminada and Amgoud, 2007] the authors propose three properties for a semantics used on ASPIC frameworks - the so called rationality postulates. Their objective is to ensure certain logical standards for reasoning with ASPIC. Thus, their main concern are the conclusions derived from the accepted arguments.

Definition 6.10. (conclusions) Let $(P, S, D(n))$ be a tuple of of premises, strict and def. rules with a naming function and let $F = (A, R)$ be the resulting ASPIC framework. For a semantics ς we define the set of conclusions from some $E \in \varsigma(F)$ to be $Concs(E) = \{Conc(x) \mid x \in E\}$.

The conclusions themselves are logical formulas again, so we need some logical background first.

Definition 6.11. (closure) Let $P \subseteq L$ be a set of formulas and S be a set of (instantiated) strict rules. The closure $Cl_S(P)$ of P under S is recursively defined as follows:

1. $P \subseteq Cl_S(P)$
2. If $s = (p_1, ..., p_n \to c) \in S$ and $p_1, ..., p_n \in Cl_S(P)$ then $c \in Cl_S(P)$.

P is closed under S iff $P = Cl_S(P)$.

Definition 6.12. (consistency) Let $P \subseteq L$, S a set of strict rules. P is consistent iff $p \neq -q$ for all $p, q \in P$.
S is consistent w.r.t. P iff P is indirectly consistent w.r.t. S iff $Cl_S(P)$ is consistent.

The rationality postulates themselves now simply apply these two principles to the set of conclusions.

Definition 6.13. (rationality postulates) Let ς be a semantics and $F = (A, R)$ the ASPIC framework for some given $(P, S, D(n))$.

1. ς satisfies Direct Consistency iff $Concs(E)$ is consistent for any extension $E \in \varsigma(F)$ and for $E = \bigcap_{D \in \varsigma(F)} D$ the set of skeptically accepted arguments.

2. ς satisfies Closure iff $Concs(E) = Cl_S(Concs(E))$ for all $E \in \varsigma(F)$ and for $E = \bigcap_{D \in \varsigma(F)} D$.

3. ς satisfies Indirect Consistency iff $Cl_S(Concs(E))$ is consistent for all $E \in \varsigma(F)$ and for $E = \bigcap_{D \in \varsigma(F)} D$.

The ideal semantics for ASPIC according to the rationality postulates generates consistent conclusions which are closed under strict rules and produce only consistent formulas under them. From a logical perspective asking for these properties makes a lot of sense. The representation of formulas with ASPIC and reasoning on it with argumentation semantics amounts to applying a formal semantics to a given logical syntax and of course we want a semantics satisfying soundness(consistency) as well as completeness(closure). So in order to adequatly represent defeasible reasoning with ASPIC frameworks the rationality postulates have to be respected. Note that both forms of consistency usually require the set S to be consistent in the first place, because strict arguments cannot attack each other in ASPIC. We include some simple consequences of the postulates for later reference:

Proposition 6.14. *Let ς be a semantics and $F = (A, R)$ the ASPIC framework for some given $(P, S, D(n))$.*

I. If $Concs(E)$ is consistent\closed\indirectly consistent for all $E \in \varsigma(F)$ then $Concs(E)$ is directly consistent\closed\indirectly consistent for $E = \bigcap_{D\in\varsigma(F)} D$.

II. If ς satisfies Direct Consistency and Closure, it also satisfies Indirect Consistency.

III. If ς is a conflictfree semantics and S is consistent w.r.t. P, ς satisfies Direct Consistency.[6]

In [Caminada and Amgoud, 2007] it is shown, that none of the standard Dung-style semantics satisfy Closure and Indirect Consistency in ASPIC$^-$. We will demosntrate this with the following counterexample for ad^c and most of the other semantics examined in this work.[7]

Example 6.15. Consider again Example 6.9. The argument set $E = \{[q], [[q] \Rightarrow t], [[q] \Rightarrow p]\}$ is a c-,cog-,naa-,lub- and nua-preferred extension (and complete and admissible for all these semantics as well as grounded for naa-, nua- and cog-semantics). It does however not satisfy Closure, because the conclusion $\neg p$ of the strict rule $p \rightarrow \neg p$ is not included in $Concs(E)$ while p is. And since $\neg p = -p \in Cl_S(Concs(E))$, E also violates Indirect Consistency.

For the c-grounded semantics a simple additional condition can be added to guarantee Closure and thus by Prop. 6.14 (II) also Indirect Consistency.

[6] The proof for these statements can be found in [Caminada and Amgoud, 2007]

[7] For a counterexample for $gr^c(= gr^{lub})$ see Example 4 in [Caminada and Amgoud, 2007]

Proposition 6.16. *Given some $(P, S, D(n))$ the c-grounded extension of the resulting ASPIC^{-} framework $F = (A, R)$ satisfies all three rationality postulates if S is consistent w.r.t. P and closed under transpositions that is for any strict rule $s = (p_1, ..., p_n \to c) \in S$ the set of its transpositions*

$$T_s := \{t_i = (\neg c, p_1, ..., p_{i-1}, p_{i+1}, ..., p_n \to \neg p_i) \mid 1 \leq i \leq n\}$$

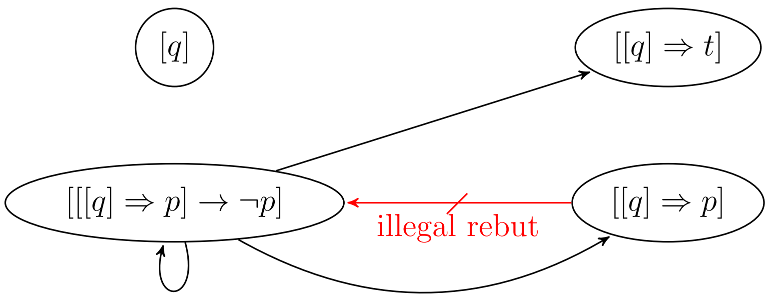
is included in the set of strict rules ($T_s \subseteq S$).

The solution proposed for c-semantics in general was to limit attacks to the direct conclusions of defeasible rules. The idea is for certain arguments to be no longer able to c-defend themselves. This was accomplished by refining the notion of Rebut.

Definition 6.17 (Restricted Rebut)**.** Let $(P, S, D(n))$ be the respective sets of premises, strict and def. rules and let A be the resulting set of arguments. An argument $x \in A$ attacks an argument $y \in A$, $(x, y) \in R$ iff y is a defeasible argument and either x undercuts y or there exist a subargument of y with a defeasible top-rule $y' = (p_1, ..., p_n \Rightarrow c) \in Sub(y)$ and a subargument $x' \in Sub(x)$ such that $Conc(x') = -Conc(y')$ then x *restrictedly rebuts* y.

So Rebut is further to only applied directly on an instance of a defeasible rule. Let us apply this modification to our running example.

Example 6.18. Let $(P, S, D(n))$ be the same as in Example 6.9. Then the attack from $[[q] \Rightarrow p]$ to $[[[q] \Rightarrow p] \to \neg p]$ is no longer allowed under Restricted Rebut.



This restriction was adapted into the - by now widely acknowledged - ASPIC^{+} formalism, which succeeded ASPIC^{-}. While it works nicely for c-semantics to reduce the number of attacks, Restricted Rebut does not solve any part of the problem for weak semantics as Example 6.18 shows. Even if the attack from $[[q] \Rightarrow p]$ to $[[[q] \Rightarrow p] \to \neg p]$ is deleted, $[[q] \Rightarrow p]$ is still accepted under naa- and any of the other weak semantics, because its only attacker is a self-attacker. It becomes clear that, in order to accept $[q] \Rightarrow t$ but not $[q] \Rightarrow p$, we need a different approach to the problem of indirect inconsistency in extensions for weak semantics.

6.3 Deductive ASPIC$^-$

One part of the idea to realize closure for e.g. complete semantics in ASPIC$^+$ was to close the set S of strict rules under transposition. This has only a limited effect, since those transposition rules may not become arguments due to missing premises. In [Cramer and Bhadra, 2020] a more direct approach is chosen. Its central objective is to represent the indirect conflicts between formulas in the ASPIC framework. As a first step the logical requirements the postulates make for conclusions are translated into direct requirements for the arguments accepted by a semantics.

Definition 6.19 (deductive semantics)**.** A semantics ς is said to be *deductive* iff for the ASPIC$^-$ framework $F = (A, R)$ of any $(P, S, D(n))$ and for any $E \in \varsigma(F)$:

If $x = [x_1, ..., x_n \to c] \in A$ is an argument with strict top-rule and $x_i \in E$ for all $i \in (1, ..., n)$ then $x \in E$.

The deductive property can be understood as Closure on argument level. If the arguments for the premises of a strict rule are part of an extension, the argument implementing the strict rule must be part of the extension too. It was shown in [Cramer and Bhadra, 2020] that for a deductive semantics the rationality postulates follow automatically for consistent sets of strict rules.

Proposition 6.20. *Let ς be an argumentation semantics. If ς is deductive then ς satisfies Closure on the ASPIC$^-$ framework $F = (A, R)$ of any $(P, S, D(n))$ and in case the set of strict rules S is consistent w.r.t. P, it also satisfies Direct and Indirect Consistency.*

Proof. Def. 6.3 guarantees that from any strict rule $s = (p_1, ...p_n \to c) \in S$ for which arguments with the premises $p_1, ..., p_n$ as conclusions exist an argument x_s with s as its top-rule exists. So if some $E \in \varsigma(F)$ violates Closure then there exists a rule $s \in S$ and argument x_s such that x_s is not included in E, so ς is not deductive. The second statement follows from the first and Prop. 6.14(I,II). □

A deductive semantics does not accept arguments if it rejects an argument strictly derived from them. With the ASPIC formalism as it is, none of our weak semantics satisfies this, remember Example 6.9, where the self-attacker is a strict consequence of an argument it attacks. The second step towards integrating inconsistencies is the introduction of additional arguments. They directly represent the relevant transpositions and apply the consequences of conflicts among strictly inferenced arguments backwards through the attack relation on the arguments they were derived from.[8]

Definition 6.21 (deductive completion)**.** Let $F = (A, R)$ be the ASPIC$^-$ framework for some given $(P, S, D(n))$. The deductive completion[9] $DC(F)$ of F is constructed according to the following rules.

(A) For each argument $x = [x' \to c] \in A$ having a strict top-rule with a single premise the argument $dc(x)$ and the attacks $(dc(x), x')$, $(x, dc(x))$ are added to F in $DC(F)$.

(B) For each argument $x = [x_1, ..., x_n \to c] \in A$ with a strict top-rule with multiple premises the arguments $dc_i(x), tc_i(x)$ for every $i \in (1, ..., n)$ are added to F in $DC(F)$, and for each i the following attacks:

1) $(x, tc_i(x))$ (conclusion attacks transposition)
2) $(tc_i(x), x_i)$ (transposition attacks premise)
3) $(x_i, dc_i(x))$ (premise attacks negated premise)
4) $(dc_i(x), tc_j(x))$ for every $j \in (1, ..., n), j \neq i$ (negated premise attacks any other transposition)

The deductive completion succeeds at establishing a link between an argument and the arguments that strictly depend on it. It does so by putting the dc-argument between them, which can be read as "suppose the conclusion c of x is wrong, then the conclusion of x' is wrong, too". It therefore attacks x' and is in turn attacked by argument x, which states named c as its conclusion. The same idea, namely transpositions, is applied to arguments with more then one premise, where we need a dc-argument for each premise and the pc-arguments in order to adequatly simulate transpositions like $\neg c_x, c_{x_1}, ..., c_{x_n} \to \neg c_{x_i}$.

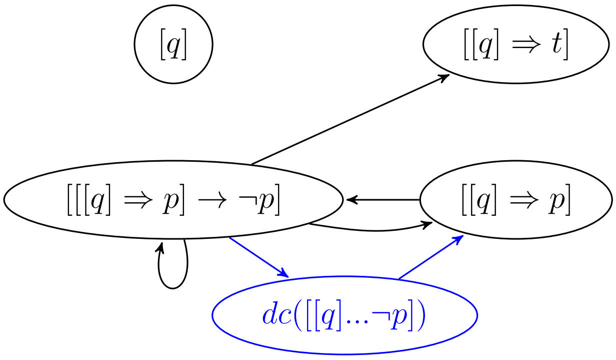
Note that the additional attacks in $DC(F)$ do not fall under the categories rebut and undercut and may by definition attack strict arguments as well. Far from causing problems this could even make it possible to satisfy the rationality postulates with an inconsistent set S of strict rules,[10] albeit some strict arguments might become unacceptable this way. Let us check how this concept changes the situation for our running example.

[8]The original idea in [Cramer and Bhadra, 2020] is to first introduce a support relation for cases of deductive dependencies and then transform it into additional arguments, but we want to focus on the effect of the additional arguments here.

[9]The name deductive completion was chosen because the original $flat(F)$ is too much out of context here.

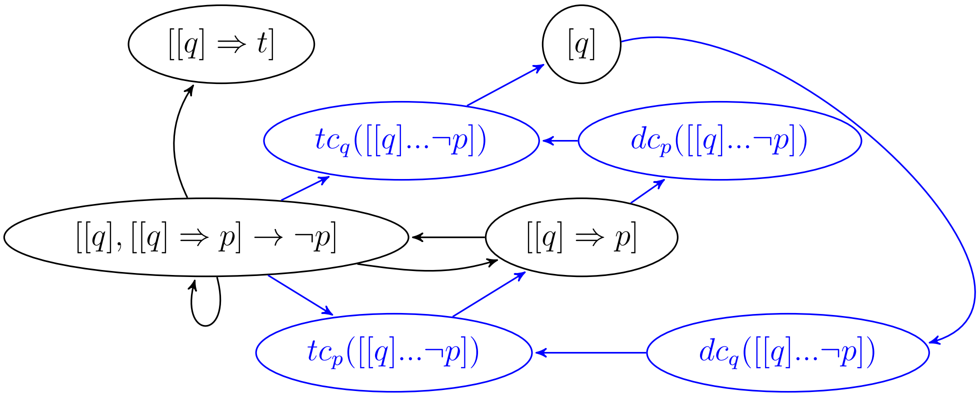
[10]hypothesis, proof for this is not contained in [Cramer and Bhadra, 2020]

Example 6.22. The deductive completion of Example 6.9 is



Before we get started on semantics for the deductive completion, we provide a slightly modified version of our example. This time a strict rule with two premises is included.

Example 6.23. Let $P, D(n)$ as in Example 6.9 but $S' = \{q, p \to \neg p\}$. The deductive completion of the ASPIC$^-$ framework of $(P, S', D(n))$ then is



The last step comes as no surprise - we apply our argumentation semantics on the deductive completion of our ASPIC framework instead. Afterwards we simply accept those arguments of the original AF which are part of an extension from the deductive completion.

Definition 6.24. Let ς be a semantics, $F = (A, R)$ the ASPIC$^-$ framework for some $(P, S, D(n))$. The corresponding dc-semantics

$$sup_\varsigma(F) := \{E \cap A | E \in \varsigma(DC(F))\}$$

takes the ς-extensions of the deductive completion $DC(F)$ and returns their intersections with the original ASPIC$^-$ framework F.

Applying this definition on our weak semantics now finally yields some results w.r.t. the rationality postulates. For c-semantics, of course, the desired results are already demonstrated in [Cramer and Bhadra, 2020] and for cogent semantics the deductiveness of the corresponding semantics follows analagously.

Proposition 6.25. *sup_{ad^c} and $sup_{ad^{cog}}$ are deductive.*

Proof. The proof for sup_{ad^c} is given in [Cramer and Bhadra, 2020]. Now let $F = (A, R)$ be the ASPIC$^-$ framework of some $(P, S, D(n))$. Since the additional arguments are no self-attackers we can argue analogously to the proof for classic semantics. An extension of the deductive completion $E \in ad^{cog}(DC(F))$ containing x' for some argument $x = [x' \to c]$ must defeat $dc(x)$ in order to be cogent. But since x is the only attacker of $dc(x)$ and E only defeats E^+ and self-attackers, it follows that $x \in E$.
For arguments with multiple premises $x = [x_1, ..., x_n \to c]$ an extension with $x_1, ..., x_n \in E$ must defeat all the transposition arguments $tc_i(x)$ which are attacked only by x and $dc_j(x)$ for $j \neq i$. But the $dc_j(x)$ are attacked by the x_j themselves, so $dc_j(x) \notin E$ for all j and thus tc_i is not defeated by some dc_j. The only remaining attacker is x, so $x \in E$.
Since x is an argument of the original framework F, $x \in E \cap A$ so $x \in_{ext} sup_{ad^{cog}}(F)$. □

Because naa-defeat rules out arguments attacked by unattacked arguments the naa-admissible semantics is too weak to be deductive, not all defended arguments have to be included after all. But the naa-complete semantics and its two subsets work with deductive completion.

Proposition 6.26. *$sup_{pref^{naa}}$, $sup_{co^{naa}}$ and $sup_{gr^{naa}}$ are deductive, $sup_{ad^{naa}}$ is not.*

Proof. Let $(P, S, D(n))$ be some sets of premises, strict and def. rules, $F = (A, R)$ the corresponding ASPIC$^-$ framework and $E \in pref^{naa}DC(F)$.

(Case 1) Suppose $x' \in E$ for some $x = [x' \to c] \in A$. Then $dc(x)$ is naa-defeated by E so either $E \to dc(x)$ or $dc(x) \notin_{ext} ad^{naa}(DC(F)^E)$. The only attacker of $dc(x)$ is x, so if $E \to dc(x)$ then $x \in E$.
Now suppose $dc(x) \notin_{ext} ad^{naa}(DC(F)^E)$. Since $dc(x)$ is no self-attacker by definition, it follows that for its only attacker, x, there exists a $D \in ad^{naa}(DC(F)^{E^{\{dc(x)\}}})$ with $x \in D$. $dc(x)$ attacks only x' so the set $DC(F)^E \setminus \{dc(x)\} = DC(F)^{E^{\{dc(x)\}}}$ is unattacked in $DC(F)^E$. By directionality of $pref^{naa}$[11] it follows that $D \in ad^{naa}(DC(F)^E)$. Then because of modularization $D \cup E \in ad^{naa}(DC(F))$. But that contradicts the maximality of E because we assumed E is naa-preferred. So D is already a subset of E and $dc(x) \in E^+$ to begin with.

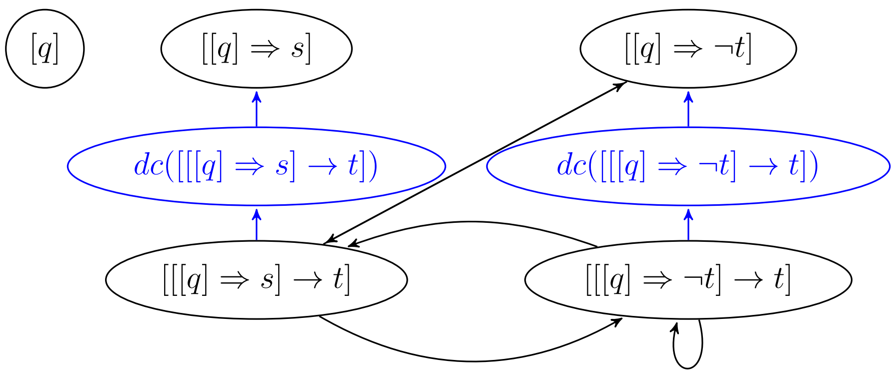
[11] see [Baumann et al., 2020a]

(Case 2) Suppose $x = [x_1, ..., x_n \to c] \in A$ and $x_i \in E$ for every $i \in (1, ..., n)$. Then $dc_i(x) \notin DC(F)^E$ for all $i \in (1, ..., n)$, because $x_i \to dc_i(x)$. So the only remaining attacker for the transposition arguments $tc_i(x)$ is x. For E to defeat $tc_i(x)$ now either $tc_i(x) \in E^+$ is satisfied, then $x \in E$, or $tc_i(x) \notin_{ext} ad^{naa}(DC(F)^E)$ which leads to the same contradiction with the maximality of E as in Case 1.

Since x is an argument of the original framework F in both cases we have $x \in E \cap A$, so $x \in_{ext} sup_{pref^{naa}}(F)$.

For $E \in co^{naa}(DC(F))$ we can argue that in both cases x is the only attacker left to make $dc(x)$ (resp. $tc_i(x)$) not naa-admissible in the reduct and that $DC(F)^{E^{\{dc(x)\}}}$ (resp. $DC(F)^{E^{\{tc_i(x) \mid i \in (1,...,n)\}}}$) is unattacked in $DC(F)^E$. Now if x was attacked by some naa-admissible extension $D \in ad^{naa}(DC(F)^{E^{\{dc(x)\}}})$, then because $dc(x)$ is only attacked by $x \in D^+$ it follows that $D \cup \{dc(x)\} \in ad^{naa}(DC(F)^E)$ so $dc(x)$ would be a legal attacker of E. Therefore no attacker of x in the reduct is naa-admissible, so $x \in \chi_{naa}(E)$. But then E is not naa-complete unless $x \in E$. The same applies to Case 2.
Since $gr^{naa}(DC(F)) \subseteq co^{naa}(DC(F))$ the proof for $sup_{co^{naa}}$ applies to $sup_{gr^{naa}}$, too. For $sup_{ad^{naa}}$ consider the following counterexample. □

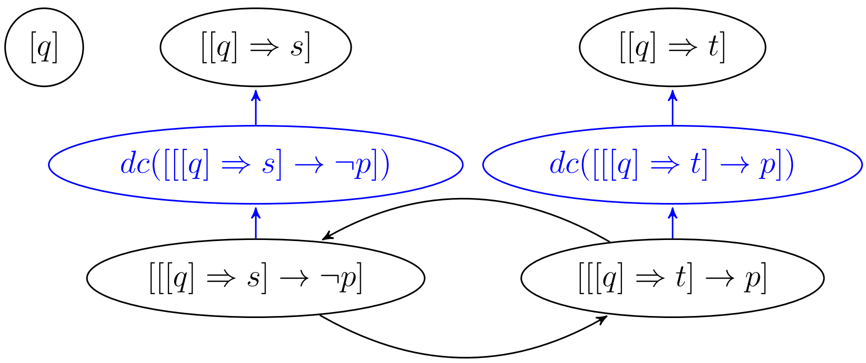
Counterexample 6.27. Let $P = \{q\}$, $S = \{s \to t, \neg t \to t\}$, $D(n) = \{q \Rightarrow s, q \Rightarrow \neg t\}$ with $DC(F)$ for the corresponding $F = (A, R)$ as below. Then $E = \{[[q] \Rightarrow s]\}$ is naa-admissible but does not include $[[[q] \Rightarrow s] \to t]$.



Lub-semantics and nua-semantics on the other hand are too weak to be deductive. The lub-grounded semantics is a special case, of course, because it coincides with the c-grounded semantics which is deductive. It is also likely that the nua-grounded semantics is deductive, although we cannot present a proof for this yet. Our reasoning is that nua-defeat defeats a lot of arguments so minimal nua-complete extension do not have to include that much.

Proposition 6.28. *sup_ς is not deductive for $\varsigma \in \{ad^{lub}, co^{lub}, pref^{lub}, ad^{nua}, co^{nua}, pref^{nua}\}$.*

Proof. Let $P = \{q\}$, $S = \{s \to \neg p, t \to p\}$, $D(n) = \{q \Rightarrow s, q \Rightarrow t\}$ with the deductive completion $DC(F)$ of the corresponding ASPIC$^-$ framework as below.[12] Then $E = \{[q], [[q] \Rightarrow s], [[q] \Rightarrow t]\}$ is an extension of $\varsigma(DC(F))$ for all the semantics in Prop. 6.28 but does not contain e.g. $[[[q] \Rightarrow s] \to \neg p]$, so none of the above semantics are deductive.



□

Conjecture 6.29. *$sup_{gr^{nua}}$ is deductive.*[13]

We now have an ASPIC formalism, namely *Deductive ASPIC$^-$*, for which two of our four weak semantics are deductive and thus by Prop. 6.20 satisfy the rationality postulates.

Corollary 6.30. *For $\varsigma \in \{ad^{cog}, co^{cog}, gr^{cog}, pref^{cog}, co^{naa}, gr^{naa}, pref^{naa}\}$ it holds that sup_ς satisfies Closure on the ASPIC$^-$ framework $F = (A, R)$ of any $(P, S, D(n))$ and in case the set of strict rules S is consistent w.r.t. P, it also satisfies Direct and Indirect Consistency.*

The direct implementation of deductive dependencies in the argument structure in *Deductive ASPIC$^-$* works in favor of thinking in terms of weak admissibility. It addresses the issue of indirect inconsistency more thoroughly than the transposition solution in [Caminada and Amgoud, 2007] by ensuring that inconsistencies between arguments have an impact on the defense of their premise arguments as well. Furthermore, the clear distinction between the original AF and its deductive completion simplifies generating only the conclusions asked for compared to transpositions within the given rule sets. In addition, the logical relations are reflected

[12] This example is analogous to Example 4 in [Caminada and Amgoud, 2007]

[13] This is based on the observation that having an nua-admissible attacker in the reduct often means an extension also has an nua-admissible attacker in the overall AF, so the empty set defeats such an extension.

more clearly in the arguments with *Deductive* $ASPIC^-$, making it a valuable graphic representation. To conclude, while introducing a support relation might seem appaling at first (and is not necessary for the framework behind it to work, as has been shown here), the issue of representing indirect inconsistencies in ASPIC is solved by this approach in a direct and reasonable manner that opens the ASPIC formalism to other semantics beside the classic Dungstyle-semantics.

Chapter 7

Conclusion

7.1 On principles for weak semantics

[Baroni and Giacomin, 2007] and [Caminada and Amgoud, 2007] have a great influence on the field of argumentation because they propose useful standards an argumentation semantics should live up to. Weak semantics violate some of these principles on purpose, others, like I-maximality, are not relevant for the defense behavior of a semantics. Nonetheless, the analysis of weak semantics under known principles conducted in [Dondio and Longo, 2021], [Dauphin et al., 2020] and [Baumann et al., 2020a] contributed greatly to this work. While we do conduct a short principle-based analysis for our newly introduced nua-semantics, a lot of open questions remain about the properties of nua-semantics and the other new semantics our defense notion yielded, e.g. co^{cog} or ad^{lub}. We decided to focus on principles which are specialized for weak semantics in this work, but conducting a standard analysis for nua- as well as cogent semantics would be valuable to conclusively compare those two approaches with lub- and naa-semantics.

A number of principles particularly suited for weak semantics are introduced in [Baumann et al., 2020a], notably modularization and unattack inclusion. We extend these in Chapter 5 by introducing e.g. c-grounded inclusion or the separation property. For future research on weak semantics a generalized modularization or other types of partitioning might help to break down the complexity of structure-sensitive defeat operators like naa-defeat, the single-additivity we adapt from c-semantics in Section 4.4 is an extreme case of such a partitioning. Two other structure oriented properties we introduce are the defeat criterion by added attacks and duplication. The two concepts are related to works on attack refinement [Boella et al., 2009] and expansion operators [Bistarelli et al., 2018] respectively. Future developments of the two principles should be discussed in this context, e.g. duplication could be applied to Sub-AFs too. The sensitivity or insensitivity, of weak semantics in particular, to certain structural changes needs further investigation. Dondio&Longo discuss two, more philosophical, principles in their works, *in dubio pro reo* and *beyond reasonable doubt*. We take up the former as our motivation for default acceptance. These three principles focus on practical, intuitive argumentation, while the others are more technical. If weak semantics are to become relevant in practice, we will have to pay attention to both.

Under our general defense notion a lot of properties regarding the extensions themselves like I-maximality can be guaranteed independently from defeat by choosing the right subset of admissible extensions as a semantics. We are now in a position to separate the study of such properties from the study of defense-related criteria. For a better understanding of the new concept we have generalized several classic properties discussed in [Dung, 1995] for arbitrary defeat operators. We have discussed monotony, the Fundamental Lemma and the direct comparison of defeat operators with conclusive results. Two aspects of c-semantics remain on which we need to shed some light in the future. The first question is under which conditions the grounded semantics is unique and the second, more important one, is under which conditions the preferred extensions are the maximal complete extensions. The generalized Fundamental Lemma is one such condition but since naa-semantics, which violate the Lemma, satisfy this, too, it is obviously too strong.

The properties adapted or newly introduced in Chapters 4 and 5 are steps towards thinking argumentation syntax and semantics together. The expressiveness of Dung-style argumentation depends on how good we make use of their combination. This especially applies to structured argumentation formalisms like ASPIC [Modgil and Prakken, 2014]. The proper embedding of an argumentation semantics into such a formalism proved difficult multiple times in the past with our Chapter 6 being a good example of this struggle.

7.2 On generalizing defense

The original *defense by attack* introduced in [Dung, 1995] is too limited in its expressiveness for most applications of abstract argumentation. A variety of solutions for this problem already exists, the majority of them can be divided into two different approaches - expanding the syntax or changing the semantics. Expanding the syntax amounts to changing the underlying argumentation framework. In order to introduce a defeat notion which takes the status of the attacker into account, for example, [Martınez et al., 2006] replace the attack relation with a conflict and a preference relation. Or instead of defining Defeat for sets of arguments on an abstract level like it has been done here, attacks can be defined for sets or arguments to begin with, resulting in SETAFs [Dvořák et al., 2019]. In the same sense *Deductive ASPIC*$^{-}$ [Cramer and Bhadra, 2020] boosts the expressiveness of ASPIC by adding the joint support relation, although the modified AF is broken down to a standard AF for the evaluation again. Another interesting proposal along these lines is made by [Hanh et al., 2010] - attacks on attacks. It might be worth it to adapt this concept on a more abstract level for our defense notion in the future. If we defeat attacks instead of arguments we could circumvent Theorem 3.10 and extend the range of our general defense notion to other types of semantics like non-conflictfree semantics.

The second option - to change the semantics - has brought us the various weak semantics already mentioned in the introduction and the chapter Previous Work and the new defense notion follows this approach as well. The idea to bring about the necessary generalization is the introduction of defeat as a flexibly definable set version of attacks. A more logic-oriented analysis on how to represent set-based defeat in Dung-style framworks can be found in [Verheij, 1995]. The defeat-based defense notion from Chapter 3 allows for a great variability in both the adapted defense concepts and the type of admissibility-based semantics generated while maintaining most of the classic formalism (defense operator, fixpoint-completeness, conflict-freeness). Now that the formal groundwork for deriving a non-classic semantics family from a single operator has been laid, future semantics designs can focus on realizing certain defense or defeat principles without having to worry about a working algorithm for generating extensions. As demonstrated this approach also enables us to directly compare the defense of semantics with fundamentally different designs. In regard of this, a generalization of the concept of strong admissibility and/or a defeat operator for SCC-semantics like the c-grounded semantics and the ub-semantics from [Dondio and Longo, 2021] is the next major challenge for our defense concept. The possibility of formulating a defeat operator for the cyclically cogent semantics proposed in [Bodanza and Tohmé, 2009] should also be examined. In general the problem of cycle-homogenity i.e. the equal treatment of even and odd cycles deserves more attention in the context of weak semantics.

The essence of the semantics approach is to take the overall situation of an AF into account when deciding on the acceptability of certain arguments. As a consequence the evaluation process can become very complex, leading to a higher computational complexity compared to classic semantics (see [Dvorák et al., 2020] on the complexity of naa-semantics for instance). On the bright side we need no additional information for this approach, we still work with the simple basic structure of arguments and an attack relation. In contrast modifications on the syntactic level need far more specifications about the arguments and their relations but applying the classic semantics on them may turn out more efficient then applying a weak semantics to a standard AF, depending on the concrete semantics and framework, of course. Future Applications of abstract argumentation will have to combine both approaches based on the information structure available, the relevant argumentation principles and the affordable computing power. Towards this end we ought to take a better look at what we already have in argumentation instead of what we want. The meta-analysis and systematic categorization of existing argumentation semantics will ultimately bring future research closer together again and ease the adaption of new ideas into existing solutions. Generalizing defense is but one step towards structuring known approaches and developing a unifying meta-theory for abstract argumentation.

Bibliography

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

[Baroni et al., 2011] Baroni, P., Caminada, M., and Giacomin, M. (2011). An introduction to argumentation semantics. *The knowledge engineering review*, 26(4):365–410.

[Baroni and Giacomin, 2007] Baroni, P. and Giacomin, M. (2007). On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700.

[Baumann et al., 2020a] Baumann, R., Brewka, G., and Ulbricht, M. (2020a). Comparing Weak Admissibility Semantics to their Dung-style Counterparts – Reduct, Modularization, and Strong Equivalence in Abstract Argumentation. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, pages 79–88.

[Baumann et al., 2020b] Baumann, R., Brewka, G., and Ulbricht, M. (2020b). Revisiting the foundations of abstract argumentation – semantics based on weak admissibility and weak defense. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2742–2749.

[Bistarelli et al., 2018] Bistarelli, S., Santini, F., and Taticchi, C. (2018). On looking for invariant operators in argumentation semantics. In *The Thirty-First International Flairs Conference*.

[Bodanza and Tohmé, 2009] Bodanza, G. A. and Tohmé, F. A. (2009). Two approaches to the problems of self-attacking arguments and general odd-length cycles of attack. *Journal of Applied Logic*, 7(4):403–420. Special Issue: Formal Models of Belief Change in Rational Agents.

[Bodanza et al., 2014] Bodanza, G. A., Tohmé, F. A., and Simari, G. R. (2014). Beyond admissibility: accepting cycles in argumentation with game protocols for cogency criteria. *Journal of Logic and Computation*, 26(4):1235–1255.

[Boella et al., 2009] Boella, G., Kaci, S., and Van Der Torre, L. (2009). Dynamics in argumentation with single extensions: attack refinement and the grounded extension. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1213–1214.

[Bromuri and Morge, 2013] Bromuri, S. and Morge, M. (2013). Multiparty argumentation game for consensual expansion applied to evidence based medicine. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 33–37. Springer.

[Calegari et al., 2019] Calegari, R., Contissa, G., Lagioia, F., Omicini, A., and Sartor, G. (2019). Defeasible systems in legal reasoning: A comparative assessment. In *JURIX*, pages 169–174.

[Caminada and Amgoud, 2007] Caminada, M. and Amgoud, L. (2007). On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310.

[Carrera and Iglesias, 2015] Carrera, Á. and Iglesias, C. A. (2015). A systematic review of argumentation techniques for multi-agent systems research. *Artificial Intelligence Review*, 44(4):509–535.

[Cramer and Bhadra, 2020] Cramer, M. and Bhadra, M. (2020). Technical report of" deductive joint support for rational unrestricted rebuttal". *arXiv preprint arXiv:2005.03620*.

[Dauphin et al., 2021] Dauphin, J., Rienstra, T., and der Torre, L. v. (2021). New weak admissibility semantics for abstract argumentation. In *International Conference on Logic and Argumentation*, pages 112–126. Springer.

[Dauphin et al., 2020] Dauphin, J., Rienstra, T., and Van Der Torre, L. (2020). A principle-based analysis of weakly admissible semantics. In *Computational Models of Argument*, pages 167–178. IOS Press.

[Dondio and Longo, 2018] Dondio, P. and Longo, L. (2018). A proposal to embed the in dubio pro reo principle into abstract argumentation semantics based on topological ordering and undecidedness propagation. In *AI^3@ AI* IA*, pages 42–56.

[Dondio and Longo, 2021] Dondio, P. and Longo, L. (2021). Weakly complete semantics based on undecidedness blocking.

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

[Dvořák et al., 2019] Dvořák, W., Fandinno, J., and Woltran, S. (2019). On the expressive power of collective attacks. *Argument & Computation*, 10(2):191–230.

[Dvorák et al., 2020] Dvorák, W., Ulbricht, M., and Woltran, S. (2020). Weak admissibility is pspace-complete. In *NMR 2020 Workshop Notes*, page 100.

[Fox et al., 2007] Fox, J., Glasspool, D., Grecu, D., Modgil, S., South, M., and Patkar, V. (2007). Argumentation-based inference and decision making–a medical perspective. *IEEE intelligent systems*, 22(6):34–41.

[Hanh et al., 2010] Hanh, D. D., Dung, P. M., Hung, N. D., and Thang, P. M. (2010). Inductive Defense for Sceptical Semantics of Extended Argumentation. *Journal of Logic and Computation*, 21(2):307–349.

[Martınez et al., 2006] Martınez, D. C., Garcıa, A. J., and Simari, G. R. (2006). On acceptability in abstract argumentation frameworks with an extended defeat relation. *Frontiers in Artificial Intelligence and Applications*, page 273.

[Modgil and Prakken, 2014] Modgil, S. and Prakken, H. (2014). The aspic+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62.

[Prakken, 2011] Prakken, H. (2011). An overview of formal models of argumentation and their application in philosophy. *Studies in logic*, 4(1):65–86.

[Tarski, 1955] Tarski, A. (1955). A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of Mathematics*, 5(2):285–309.

[Verheij, 1995] Verheij, B. (1995). Accrual of arguments in defeasible argumentation. In *Proceedings of the Second Dutch/German Workshop on Nonmonotonic Reasoning*, pages 217–224. Citeseer.